

Advanced Anomaly Detection and Quality Control in PCB Manufacturing

Marzieh Hashemzadeh Saadat

**A Thesis
in
The Department
of
Institute for Information Systems Engineering**

**Presented in Partial Fulfillment of the Requirements
for the Degree of
Master of Applied Science (Quality Systems Engineering) at
Concordia University
Montréal, Québec, Canada**

December 2024

© Marzieh Hashemzadeh Saadat, 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Marzieh Hashemzadeh Saadat**

Entitled: **Advanced Anomaly Detection and Quality Control in PCB Manufacturing**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

Dr. Mohsen Ghafouri Chair

Dr. Mohsen Ghafouri External Examiner

Dr. Manar Amayri Examiner

Dr. Farnoosh Naderkhani Supervisor

Approved by _____
Chun Wang, Chair
Department of Institute for Information Systems Engineering

_____ 2024

Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Advanced Anomaly Detection and Quality Control in PCB Manufacturing

Marzieh Hashemzadeh Saadat

Printed Circuit Boards (PCBs) are essential in electronic devices, where even minor defects can significantly impact products and the environment. Thus, rigorous quality control is imperative in PCB manufacturing. This thesis tackles critical challenges by developing robust strategies for defect detection and accurately predicting repair needs. It begins with an extensive background on current fault detection and repair strategies. Central to this study is the use of advanced machine learning (ML) and deep learning (DL) techniques to enhance the accuracy of the PCB labeling process, integrating data from Solder Paste Inspection (SPI) and Automatic Optical Inspection (AOI) datasets. The research is structured into distinct phases, each addressing different aspects of the PCB manufacturing process. The initial phase focuses on improving the prediction of human inspection labels using advanced ML and DL techniques, particularly addressing the challenges of imbalanced datasets with synthetic data augmentation techniques like Synthetic Minority Oversampling Technique (SMOTE) and Conditional Tabular Generative Adversarial Network (CTGAN). The subsequent phase expands ML algorithms to refine the process of assigning "RepairLabel" to PCBs, incorporating ensemble methods and sophisticated feature engineering to boost accuracy and efficiency. The proposed methods have shown promising results, demonstrating their substantial potential for real-world applications. The thesis concludes with a summary of findings and discusses the implications for PCB manufacturing. It also outlines potential directions for future research, suggesting further enhancements in fault detection techniques and the development of more intelligent and efficient systems.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Farnoosh Nadarkhani, for her invaluable guidance, support, and encouragement throughout my research. Her expertise and dedication have been crucial to the successful completion of my studies. I also sincerely thank Prof. Mohsen Ghafouri and Prof. Manar Amayri for their valuable time and the great opportunity to have them as my committee members.

My heartfelt thanks are extended to my beloved spouse, whose unwavering support and understanding have been my constant source of strength. His encouragement and patience have been indispensable, and, as in all things, I am deeply grateful for his presence in my life. I am equally thankful to my parents and dear brother, whose emotional support and companionship have propelled me forward.

Lastly, I would like to acknowledge my efforts and dedication throughout this journey. The challenges and obstacles I encountered have only strengthened my resolve to succeed.

Contents

List of Figures	viii
List of Tables	ix
1 Introduction	1
1.1 Objectives and Contributions of the Study	3
1.2 Thesis Organization	5
2 Background and Literature Review	7
2.1 PCB Manufacturing and Quality Control	7
2.2 Traditional Fault Detection Techniques	8
2.2.1 visual inspection	9
2.2.2 X-ray Inspection	10
2.2.3 In-Circuit Testing	10
2.2.4 Flying Probe Testing	11
2.2.5 Boundary Scan Testing	12
2.2.6 Thermal Imaging	12
2.2.7 Ultrasonic Testing	13
2.2.8 Automated Functional Testing	13
2.3 Artificial Intelligence and Machine Learning	14
2.4 Deep Learning Techniques in PCB Defect Detection	15

3 Automating Human Operator Label in PCB Production Inspection Using ML/DL Models: The Critical Role of Synthetic Data Volume on Model Performance	17
3.1 Problem Statement	18
3.2 Dataset Overview	20
3.2.1 SPI Dataset	20
3.2.2 AOI Dataset	21
3.2.3 Descriptive Analysis of The SPI-AOI Dataset	25
3.2.4 Categorical Variables Exploration	27
3.3 Data Preparation and Pre-processing	29
3.3.1 Data Aggregation	29
3.3.2 Data Cleaning and Integrity Verification	31
3.3.3 Feature Consistency Adjustments	31
3.3.4 Feature Scaling	32
3.3.5 Data Augmentation	34
3.3.6 Data Encoding	46
3.3.7 Feature Selection	47
3.3.8 Data Splitting	48
3.4 Model Implementation	49
3.4.1 Instance-based Model	49
3.4.2 Tree-based Models	49
3.4.3 Boosting-based Models	50
3.4.4 Hybrid/Deep Learning-based Model	52
3.4.5 Tools and Libraries	54
3.5 Experimental Results	55
3.5.1 Metrics and Indicators	55
3.5.2 Results and Discussion	57
3.5.3 Comparing Our Approaches to Similar Solutions	65
3.6 Summary of The Chapter	66

4	Prediction of Human Repair Labels in PCBs: Leveraging Feature Engineering and Ensemble Learning Techniques	67
4.1	Problem Identification	68
4.2	Data Preprocessing	69
4.2.1	Data preparation and Cleaning	69
4.2.2	Data Type Standardization	70
4.2.3	Feature Engineering	71
4.2.4	Data Spilitig	76
4.3	Model Implementation	76
4.3.1	Proposed Models	77
4.3.2	Tools and Libraries	78
4.4	Evaluation and Results	78
4.4.1	Metrics	78
4.4.2	Model Training and Results	79
4.4.3	Feature importance Analysis	81
4.4.4	Comparative Review of Our Methods Against Current Works	82
4.5	Summary of The Chapter	83
5	Summary and Future Research Directions	84
5.1	Summary of Thesis Contributions	84
5.2	Future Research	85
	Appendix A Evaluation of Imbalance Techniques Using Various Synthetic Data Volumes for Operator Label Prediction	87
	Bibliography	90

List of Figures

Figure 2.1	Printed circuit board production line (Gore, Minami, Kundu, Lee, et al., 2022)	8
Figure 3.1	Common Defect Types of the PCB Board (Li, Kuo, & Guo, 2020)	19
Figure 3.2	Continuous features correlation matrix	26
Figure 3.3	Categorical Features Labels	28
Figure 3.4	Target Features Distribution	28
Figure 3.5	SMOTE Oversampling Process	36
Figure 3.6	Structure of GAN (Eom & Byeon, 2023)	37
Figure 3.7	Training-by-sampling of CTGAN (L. Xu, Skoularidou, Cuesta-Infante, & Veeramachaneni, 2019)	38
Figure 3.8	Loss values of CTGAN training process	41
Figure 3.9	Continuous Features Comparison of the generated data and actual data	43
Figure 3.10	Categorical Features Comparison of the generated data and actual data	44
Figure 3.11	TabNet Model Architecture (Arik & Pfister, 2021)	53
Figure 3.12	ROC curves for TabNet Model	64
Figure 4.1	Repair Labels Distribution	70
Figure 4.2	Feature Importance Analysis	82

List of Tables

Table 3.1	Detailed Feature Overview of the SPI Dataset	21
Table 3.2	Detailed Feature Overview of the AOI Dataset	22
Table 3.3	Statistical Summary of Features	25
Table 3.4	Synthetic Data Quality Report	43
Table 3.5	KS and TVD Metrics and Scores	45
Table 3.6	KNN Performance Metrics at Different Levels of SMOTE Oversampling . .	58
Table 3.7	KNN Performance Metrics at Different Levels of CT-GAN Synthetic Data . .	58
Table 3.8	LGBM Performance Metrics at Different Levels of SMOTE Oversampling .	59
Table 3.9	LGBM Performance Metrics at Different Levels of CT-GAN Synthetic Data .	60
Table 3.10	XGBoost Performance Metrics at Different Levels of SMOTE Oversampling	61
Table 3.11	XGBoost Performance Metrics at Different Levels of CT-GAN Synthetic Data	62
Table 3.12	TabNet Performance Metrics at Different Levels of SMOTE Oversampling .	63
Table 3.13	TabNet Performance Metrics at Different Levels of CT-GAN Synthetic Data	64
Table 3.14	F1-score Comparison for OperatorLabel Prediction	65
Table 4.1	Model Performance Results	80
Table 4.2	F1-score Comparison for RepairLabel Prediction	83
Table A.1	Cat Boost Performance Metrics at Different Levels of SMOTE Oversampling	87
Table A.2	Gradient Boosting Performance Metrics at Different Levels of SMOTE Over- sampling	88
Table A.3	Extra Trees Performance Metrics at Different Levels of SMOTE Oversampling	88

Table A.4 Random Forest Performance Metrics at Different Levels of CT-GAN Syn-	
thetic Data	89
Table A.5 Decision Tree Performance Metrics at Different Levels of CT-GAN Synthetic	
Data	89

Chapter 1

Introduction

Printed circuit boards, commonly known as PCBs are essential elements in electronic gadgets, serving as the foundation for electronic circuits. Their importance, in guaranteeing the efficiency, effectiveness, and dependability of devices, cannot be overstated ([Farkas, Géczy, Kovács, & Bonyár, 2022](#)). PCBs are ubiquitous and found in numerous applications across various industries. In the medical sector, PCBs are vital in equipment such as pacemakers, defibrillators, anesthetic machines, ECG devices, and electrosurgical units, where precise and reliable operation is critical ([Adamson et al., 2020](#)). In industrial settings, PCBs are crucial, for powering equipment like power supplies, CNC machines, and solar power systems to ensure they work efficiently. Additionally, PCBs are integral to vehicles, aircraft, and marine systems, supporting navigation systems and circuits for sensors and actuators that need to function in diverse environments ([Perdigones & Quero, 2022](#)). The consumer electronics sector, encompassing cell phones, computers, appliances, and video games, also relies heavily on PCBs. These boards play a role in meeting the demand for compact and high-performance designs that modern consumers expect ([Dervišević, Minić, Kamberović, Ćosović, & Ristić, 2013](#)). The versatile use of PCBs across various applications underscores their crucial role in modern electronics. PCBs excel in supporting circuits while ensuring top-notch performance and dependability, making them a vital component of contemporary electronic design and production. ([Worden, 2024](#)).

Given the pivotal role of PCB quality in ensuring the functionality, safety, and reliability of electronic devices, it is crucial to effectively identify defects. In today's intelligent industry landscape,

the ability to detect even minor flaws on the surface of PCBs is essential. Common manufacturing issues such as open circuits, missing components, and misplacement of parts can decrease production yield and compromise device reliability (Nguyen & Bui, 2022). Defects in PCBs can lead to device malfunctions or complete failures. High-quality PCBs that meet all specifications from design through assembly not only fulfill consumer expectations but also significantly boost a company's reputation and market share (Verna, Genta, Galetto, & Franceschini, 2023). Furthermore, the environmental and economic advantages of defect-free manufacturing are becoming increasingly important. With the rapid growth in electronic device demand over the past five decades, the scale of PCB production has expanded, contributing to a rise in electronic waste. This waste contains harmful substances that pose risks to both health and the environment (Chakraborty, Kettle, & Dahiya, 2022; C. Wu, Awasthi, Qin, Liu, & Yang, 2022). Effective quality control and early defect detection in PCB manufacturing not only ensure the production of reliable devices but also help reduce environmental pollution and promote sustainable practices (Cui & Anderson, 2016). Effective and cost-efficient inspection procedures are also critical in reducing expenses linked to quality problems. Flaws in products can impact both quality and costs, underscoring the need for businesses to enforce quality assurance practices to remain competitive. The creation of defect identification models is essential in manufacturing to anticipate defects strategize quality control and streamline production operations. These models enhance the monitoring of production, predict trends in quality fluctuations, and provide early warnings, which ultimately reduces resource wastage, optimizes yield, and minimizes losses (Verna, Puttero, Genta, & Galetto, 2023).

Traditional inspection techniques such as visual inspection, Automatic Optical Inspection (AOI), and X-ray inspection are fundamental in maintaining the quality and integrity of PCBs (Zhou, Yuan, Zhang, Ding, & Qin, 2023). Visual inspection relies on the expertise of trained personnel to identify visible defects like missing components and soldering errors, although this method is labor-intensive and prone to human error (Galetto, Verna, Genta, & Franceschini, 2020). While visual inspection depends largely on human skill and is susceptible to mistakes, the progression of inspection technologies has led to the introduction of more automated solutions such as AOI. These systems help to mitigate some of the inherent challenges associated with manual inspection. AOI systems are integral to the PCB production process, tasked with the detection and diagnosis of surface defects.

These systems utilize advanced cameras and algorithms to identify a range of issues including open circuits, absent components, and misplacements, offering increased speed and consistency in inspections. Despite their importance in ensuring product reliability, especially in the smart electronics sector, AOI systems may struggle with complex or densely populated boards. X-ray inspection provides a means to detect hidden defects such as internal solder joint issues but requires expensive equipment and specialized knowledge. Although these conventional methods are effective, they encounter challenges related to accuracy, efficiency, and scalability (Houdek & Design, 2016; Silva et al., 2019). This analysis highlights the ongoing need for innovations in PCB inspection techniques to address these challenges, ensuring high-quality electronics production in an increasingly demanding market.

As the complexity and miniaturization of PCBs advance, the limitations of conventional inspection methods highlight the need for more advanced solutions. The adoption of Artificial Intelligence (AI) and ML in detecting defects on printed circuit boards has transformed traditional quality control into more sophisticated, efficient, and reliable operations (Ural & Sezen, 2024). AI-driven methods enable rapid and precise analysis of extensive datasets, pinpointing subtle defects that might elude traditional techniques. These approaches bring multiple advantages, such as enhanced accuracy, exemplified by systems that achieve high precision in defect detection (Jun & Jung, 2023); heightened efficiency, as shown by systems that deliver efficiency coefficients above 95 percent and processing times below two seconds (Ong, Mustapha, Ibrahim, Ramli, & Eong, 2015); scalability, demonstrated by AI models that manage large data volumes and complex inspection tasks (Chaudhary, Dave, & Upla, 2017); and cost-effectiveness, through reducing the reliance on manual inspection and decreasing the occurrence of defects (Sundaram & Zeid, 2023).

1.1 Objectives and Contributions of the Study

The primary aim of this study is to thoroughly evaluate the dataset from the Prognostics and Health Management (PHM) challenge provided by Bitron Spa, a leader in the production of mechatronic devices (PHM.Society, 2022). The research specifically focuses on developing tailored ML and DL techniques to identify faults at two different stages of the PCB production line. This thesis

addresses critical challenges in data processing and model performance within defect detection systems for PCB manufacturing. It aims to provide a comprehensive evaluation of how various machine learning and deep learning strategies can significantly enhance fault detection capabilities within the manufacturing process. The significant contributions of this research are outlined as follows:

- (1) **Innovative Application of CTGAN for Imbalanced Data:** This research pioneers the use of Conditional Generative Adversarial Networks to address the challenge of imbalanced datasets within PCB manufacturing. By generating synthetic data that mimics rare defect types, CTGAN has been crucial in balancing the dataset, which enhances the model's training process and overall accuracy. This methodological innovation not only improves the reliability of fault detection but also sets a new standard for handling dataset imbalance in industrial applications.
- (2) **Synthetic Data Volume and Its Effects on Model Performance:** The key aspect of this study is the exploration of the impact of synthetic data volume on the performance of machine learning and deep learning models to identify the optimal threshold of data augmentation for our model training. This analysis provides valuable insights into how different volumes of synthetic data can optimize the learning process and model performances, contributing to more effective and efficient predictive systems.
- (3) **Feature Engineering and model enhancement:** In this study, we developed innovative feature engineering methods such as feature transformation, feature extraction and feature aggregation to enhance model predictions for PCB defect detection. These features were meticulously analyzed and integrated, resulting in significant improvements in accurately predicting defect repair labels. This contribution demonstrates the practical benefits of feature engineering in improving model outcomes and underscores the role of advanced data processing techniques in complex manufacturing environments. Our approach achieved notable enhancements in performance metrics, marking a substantial methodological advancement in the field of machine learning.
- (4) **Leveraging Ensemble Learning Strategies:** This thesis makes a significant contribution to

the field of PCB defect detection by leveraging ensemble learning techniques. Specifically, through the application of bagging, we were able to improve model stability and reduce variance, which is crucial for handling the inherently noisy data from PCB inspections. The use of these techniques, including detailed implementations of algorithms like K Nearest Neighbors (KNN) and Light Gradient Boosting Machine (LGBM), has not only optimized the defect detection process but also substantially minimized the rate of false negatives and false positives. This approach has proved to be effective in ensuring that PCBs meet the stringent quality standards required for sophisticated electronic applications, thereby significantly contributing to the enhancement of manufacturing practices and the reduction of costly production errors.

1.2 Thesis Organization

The rest of the structure of the thesis is laid out as follows:

- Chapter 2, encompasses a detailed review of existing literature relevant to the thesis topic. It also outlines the foundational concepts that will be essential for understanding the analyses and discussions in the upcoming chapters.
- Chapter 3, introduces the datasets and tools utilized in this study, detailing the critical steps in dataset preparation. This includes addressing data imbalances and implementing innovative techniques such as CTGAN. The chapter also presents experimental results from applying machine learning and deep learning models for anomaly detection in PCBs during the AOI stage.
- Chapter 4, concentrates on automating the determination of repair status for PCBs using ensemble models. This includes a specific focus on various preprocessing approaches, particularly feature engineering. This effort aims to replace manual evaluations and significantly enhance both the efficiency and sustainability of PCB manufacturing processes.
- Chapter 5, concludes the thesis by summarizing the main findings and contributions of the research. The chapter also explores potential avenues for future research in this field, suggesting

how subsequent studies could build on the presented work to further advance knowledge and understanding in the area.

Chapter 2

Background and Literature Review

2.1 PCB Manufacturing and Quality Control

This section provides a comprehensive overview of PCBs, detailing their components and the manufacturing processes. It also describes common types of faults or defects that can occur during the electronics board production line. This foundation highlights the critical issues in PCB manufacturing and underscores the necessity of exploring effective quality control techniques.

PCBs are critical components in modern electronic devices, serving as the foundation for assembling electronic circuits. Their manufacturing process involves several stages, each generating specific types of data that are crucial for defect detection and quality control. Understanding these stages and the associated data types is essential for implementing effective inspection and defect detection strategies ([Zhou et al., 2023](#)). The production line for circuit boards can vary significantly based on factors such as the board's complexity, the technology employed, and specific application requirements. Complex boards may require processes like multi-layer lamination or specialized coating procedures, whereas simpler boards may undergo a more streamlined production process. Despite these variations, the standard PCB production process generally consists of five core stages. As illustrated in [Figure 2.1](#), the production line begins with the printing machine and Solder Paste Inspection, followed by Surface Mount Device (SMD) placement. Precision in placement is essential to avoid misalignments that can lead to defects. Next, the components pass through a reflow oven, where the solder paste is melted to create solid solder joints. Then they are subjected to

Automatic Optical Inspection. The complexity and variability inherent in these production stages highlight the critical importance of robust quality control methods. Throughout these stages, various faults or defects can arise, either during the manufacturing process or due to environmental influences over time. According to (Sankar, Lakshmi, & Sankar, 2022), there are approximately 34 prevalent fault types in PCBs that are commonly focused on surface issues; problems such as soldering issues (e.g., insufficient or excessive solder) or component and joint flaws; issues such as open circuits (where connections are interrupted) and short circuits (where unintended connections occur). These defects necessitate sophisticated techniques and algorithms to accurately differentiate between normal and faulty behavior, highlighting the importance of robust quality control methods.

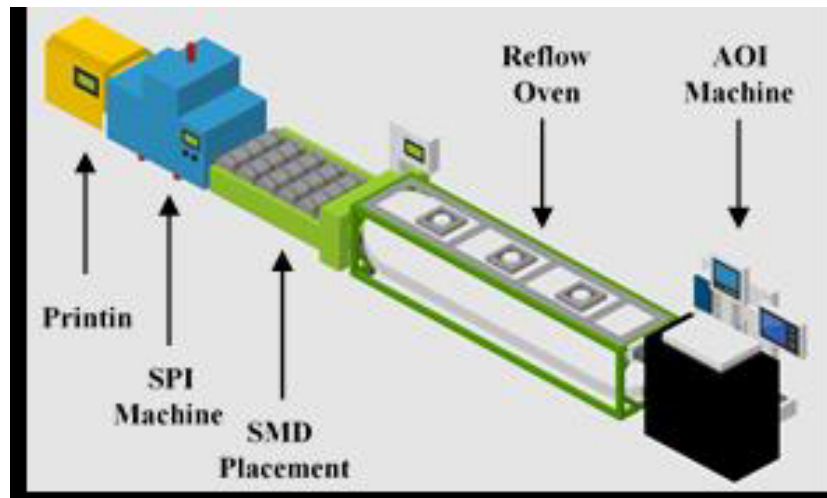


Figure 2.1: Printed circuit board production line (Gore et al., 2022)

2.2 Traditional Fault Detection Techniques

This section explains traditional fault detection methodologies in PCBs. These methodologies are divided into several primary categories, each characterized by its unique advantages and inherent limitations. By comprehensively understanding these methods, this analysis assesses their integration into existing production workflows. Furthermore, it underscores the imperative for adopting more sophisticated and automated strategies to address the dynamic requirements of the industry.

2.2.1 visual inspection

- **Human inspection**

Human inspection has traditionally played a critical role in PCB defect detection and general industrial sorting processes. In these systems, trained technicians visually examine products for defects such as soldering issues, misaligned components, and other abnormalities. This method, although fundamental, is inherently limited by human error, fatigue, and the inability to maintain high accuracy and consistency over prolonged periods. Repeated and long-term manual sorting actions are not only labor-intensive but also fail to meet the increasing demands for precision and efficiency in modern manufacturing (C. Wu, 2020). The shortcomings of manual inspection highlight the necessity of adopting AOI systems.

- **Automated Optical Inspection**

AOI leverages computer vision technology and advanced image processing algorithms to detect defects with higher accuracy and reliability. This method employs specialized cameras and algorithms to find anomalies like misplaced components or poor soldering. Unlike manual methods, AOI can handle large volumes of products, providing real-time detection and classification of defects without the limitations imposed by human factors. This automation not only improves production efficiency but also ensures a more consistent and reliable quality control process, essential for meeting the stringent standards of modern electronics manufacturing. Thus, the shift from manual visual inspection to AOI is crucial for advancing the capabilities and performance of industrial production lines (Singh, Kharche, Chauhan, & Salvi, 2024). Some studies, concentrate on surface defect detection. According to (Chauhan & Bhardwaj, 2011), the authors implemented a subtraction algorithm that compares a reference image to identify defective regions, including over-etchings, under-etchings, and holes. On the other hand, (Dai, Mujeeb, Erdt, & Sourin, 2018) presented an automatic optical inspection for Soldering Defect detection, introducing an active learning framework that starts with a small labeled subset, expanded using K-means clustering and active user input to train a Support Vector Machine (SVM) classifier. This method achieves high accuracy with minimal user input, outperforming other sampling methods and reducing annotation costs. Although

AOI inspection significantly improves the detection of surface defects, offering faster and more consistent results, it can struggle with complex PCB layouts. This limitation underscores the need for more advanced methods such as X-ray inspection, which can identify internal defects that optical systems might miss.

2.2.2 X-ray Inspection

PCB X-ray inspection involves capturing precise images of the board's interior using X-ray technology. After the PCB is set up on a platform, X-rays are emitted by an X-ray machine and penetrate the board. These X-rays are detected by a detector, which produces images that may be examined for flaws. In the next step, trained inspectors or automated systems examine these images to identify any faults and ensure the PCB's quality (Neubauer & Hanke, 1993). (Neubauer, 1997) states that this work underscores the importance of X-ray inspection in solder joint quality control. This study introduces a hierarchical approach that employs a combination of 2-D and 3-D inspection methods for fast and precise fault detection. For solder joint evaluation, Neural network-based classifiers are employed to identify anomalies. This method offers a robust solution for real-time process monitoring and defect classification.

2.2.3 In-Circuit Testing

In-Circuit Testing (ICT) is a vital methodology for assessing the integrity and functionality of PCB components by identifying potential defects in components or connections that could affect the overall performance and reliability of the board. The process involves placing the PCB on a fixture equipped with numerous probes that touch specific test points on the board. These probes deliver electrical signals to the components, and the responses are measured and compared against expected outcomes. Any discrepancies can signal issues such as incorrect component values, faulty components, or poor soldering. This testing strategy is crucial for ensuring that PCBs meet quality standards before being deployed in final products. The effectiveness of ICT has been further enhanced through the development of sophisticated software tools that automate the generation of in-circuit tests based on the product design files, even in scenarios where probe access is not available for every net. This software can also highlight areas on the PCB where fault coverage is suboptimal,

suggesting where additional probe points could enhance test quality. The application of graph-based algorithms facilitates more efficient analysis of the board's design, enabling the transformation of complex, unreachable structures into simpler, testable units while maintaining precise control over the testing environment. This advanced approach not only improves the scope and accuracy of ICT but also adapts to the evolving complexities of modern PCB designs ([Albee, 2013](#); [Houdek & Design, 2016](#)).

2.2.4 Flying Probe Testing

The previously mentioned methods did not allow for complete PCB testing due to the complex structures of PCBs and physical access. This led to the development of the automated, non-contact flying probe inspection method. In this technique, movable probes, also known as "flying probes," navigate across the PCB to test various points. Unlike traditional "bed-of-nails" testers, flying probe systems do not require custom fixtures, making them highly flexible and cost-effective for low to medium-volume production and prototype testing. These systems can precisely measure electrical parameters such as voltage and current, allowing for the detection of defects like open circuits, shorts, and incorrect component placements. The versatility and reduced setup time make flying probes particularly advantageous for educational purposes and environments with frequent design changes. For example, ([Jurj, Rotar, Opritoiu, & Vladutiu, 2020](#)) explored this method by detailing the implementation of an affordable, sensorless in-circuit tester. The tester operates on three main axes (X, Y, Z) using stepper motors and mechanical components controlled by an Arduino microcontroller. This setup allows precise measurement of voltage and current at specific test points on the PCB, enabling the detection of various defects. The sensorless approach reduces costs by eliminating the need for expensive optical sensors, instead using Cartesian coordinates to calculate the required steps for probe movement from a reference point. On the other hand, the paper ([Tsai & Huang, 2018](#)) discusses an alternative approach for PCB defect detection, focusing on Fourier image reconstruction. This method compares the Fourier spectra of a template and a test image to identify local anomalies. By retaining only the suspicious frequency components and applying the inverse Fourier transform, the technique reconstructs the test image, highlighting defects while removing the common background pattern. This Fourier-based approach is invariant to translation

and illumination, making it robust for detecting subtle defects in complex pattern surfaces.

2.2.5 Boundary Scan Testing

In the ICT method, due to the high costs of testing, there arose a need for an acceptable and cost-effective defect detection method that allows for the design of smaller PCBs. The solution of JTAG, or the Joint Test Action Group, notably leads to shrinking PCBs. It is a standardized method for testing and verifying the integrity of electronic circuits at both the chip and PCB levels. This technique utilizes a boundary scan architecture (BSA) where each pin of an IC is connected to a boundary scan cell, which can be controlled and monitored via test access ports (TAPs). This allows for the detection of structural defects such as open circuits, shorts, and incorrect placements without the need for physical probes. JTAG is highly versatile, supporting in-system programming and real-time PCB functionality monitoring. In (Paul & Bhunia, 2021), the authors use JTAG to prevent counterfeiting and tampering by leveraging the BSA to measure dynamic current variations during test pattern transmissions. This process generates unique digital signatures for each PCB and its components, which are then used to verify authenticity and detect in-field modifications. Similarly, (Shashidhara, Yellampalii, & Goudanavar, 2014) discusses using JTAG for efficient PCB defect detection. It highlights how the boundary scan method facilitates the detection of structural issues such as shorts, opens, and stuck-at faults. This is a low-overhead, high-precision approach to ensure the integrity and reliability of PCBs in various stages of their lifecycle, from manufacturing to field deployment.

2.2.6 Thermal Imaging

Another major challenge in PCB production lines is the excessive heat generated in the Integrated Circuits (ICs), which can lead to reduced performance or even failure of the PCBs. Thermal imaging involves capturing thermal images of PCBs and processing these images to detect hotspots that indicate faults helping to identify and address these thermal issues promptly. This method is effective for non-contact and non-invasive detection, offering high accuracy in distinguishing between faulty and non-faulty ICs, thus improving the reliability and safety of electronic systems. According to

([Sarawade & Charniya, 2019](#)), this efficient prototype method identifies faulty ICs with 100 % accuracy in classifying test images and requires minimal computational time. By integrating thermal cameras with the SURF algorithm, the system can alert users to circuit faults.

2.2.7 Ultrasonic Testing

Ultrasonic Testing involves a laser-induced ultrasound scanning imaging system, where a pulsed laser is utilized to excite ultrasonic waves on the PCB surface. Sensors then capture and analyze the interaction of these waves with the PCB. A key advantage of this technique is its capability to detect minute defects that traditional methods might miss, making it particularly useful for high-density PCBs where defect detection is crucial for ensuring reliability and performance. The system employs advanced signal processing techniques, such as wavelet transform denoising and principal component analysis (PCA), to enhance the signal-to-noise ratio (SNR) and accurately identify defects ([X. Chen, Tao, Shang, & Liu, 2022](#)). A comparison of this method with traditional infrared thermal wave imaging demonstrates that laser-induced ultrasound provides superior resolution and defect detection, especially for small-scale defects. This approach holds significant potential for enhancing quality control processes in PCB manufacturing, enabling early detection of potential failures and reducing the risk of defective products reaching the market ([F. Wang et al., 2022](#)).

2.2.8 Automated Functional Testing

In the evolving landscape of automotive technology, functional testing is a critical process designed to evaluate the performance of electronic modules against predefined criteria, without the necessity to delve into the internal architecture of the modules. This innovative approach leverages the LabVIEW programming environment to streamline the testing process by integrating a suite of programmable instruments, including multimeters, oscilloscopes, electronic loads, and waveform generators, all connected via the General Purpose Interface Bus (GPIB). By utilizing script files with predefined commands, the system can automatically configure these instruments, execute a series of tests, and meticulously record the results. This method excels in accurately simulating various operational scenarios and assessing the performance of electronic modules under different conditions. The automation not only drastically reduces the time and manual effort involved in repetitive

testing but also enhances the reliability and precision of the test outcomes. Furthermore, it generates comprehensive reports for in-depth analysis, thus offering a highly efficient and robust solution for functional testing in complex electronic systems ([Ana-Maria, Georgiana, & Ioan, 2018](#)).

2.3 Artificial Intelligence and Machine Learning

As the size of joints and components decreases and the number of potential flaws increases, the limitations of traditional inspection methods, including manual inspection and classical computer vision (CV), are becoming increasingly evident ([Dai, Mujeeb, Erdt, & Sourin, 2020](#)). Therefore, due to the development of modern production lines and the presence of large data sets, artificial intelligence strategies and machine learning algorithms have emerged as powerful tools for automating the defect detection process and providing substantial success in addressing problems in the field of industry. They have clear advantages in terms of cost-savings, rapid detection, and high accuracy and precision ([Putera & Ibrahim, 2010](#); [Yang et al., 2020](#)). Over time, advanced machine learning techniques such as SVM ([Zhang, Shi, Li, Zhang, & Liu, 2018](#)), Neural Networks (NN) ([Anoop, Sarath, & Kumar, 2015](#); [Ng et al., 2011](#)), Decision Trees (DT), and Genetic Algorithms (GA) ([Mashohor, Evans, & Erdogan, 2006](#)) were adopted for PCB defect detection, enhancing the sophistication of these systems. For instance, the authors ([Vafeiadis et al., 2018](#)) proposed a framework utilizing SVM with polynomial and radial basis function kernels to detect defects on PCBs. The results of the study indicate that using the full set of features maximizes classification performance, suggesting minimal advantages in reducing features for this application, needed further exploration into deep learning techniques to enhance these outcomes. In another study ([J. Chen, Zhang, & Wu, 2021](#)) developed a novel approach utilizing SVM to identify wire bonding defects in IC chips, surpassing other techniques like Vision Detection Systems (VDS) and Convolutional Neural Networks (CNN) in sensitivity, accuracy, and speed. Furthermore, integrating semi-supervised methods and advanced machine-learning techniques has also shown promise. For instance, a semi-supervised defect detection method combining SVM classifiers and K-mean clustering has been introduced for soldering anomaly detection by ([Dai et al., 2020](#)). In parallel, advancements in object detection algorithms, such as You Look Only Once (YOLO) and its derivatives YOLOV3-Mobilenet ([Huang, Gu, Sun,](#)

[Hou, & Uddin, 2019](#)), and YOLO-v5 ([Adibhatla et al., 2021](#); [Parlak & Emel, 2023](#)), have enhanced the identification of electronic components and soldering joint defect detection on circuit boards.

2.4 Deep Learning Techniques in PCB Defect Detection

The incorporation of advanced machine learning techniques has significantly enhanced the efficiency of PCB defect detection systems. As the field has progressed, deep learning methods have emerged as powerful tools to further enhance detection capabilities. This shift from conventional machine learning to deep learning represents a major advancement in accurately identifying and classifying PCB defects. Various forms of Artificial Neural Networks have been employed, including Feedforward Neural Network (FNN), Recurrent Neural Network (RNN), CNN, and Modular Neural Network (MNN) ([Gaber, Hussein, & Moness, 2021](#)). Among these neural network architectures, CNNs have gained prominence due to their superior ability to perform precise feature extraction without the need for manually defined features. This capability allows CNNs to detect subtle and complex defects more effectively. For instance, CNNs have been successfully applied in IC component defect recognition ([Lin, Wang, & Lin, 2019](#)), feature extraction, and feature fusion ([Jin et al., 2021](#)), providing robust solutions for real-time inspection in Surface Mount Technology (SMT) ([H. Wu, Lei, & Peng, 2022](#)). Further advancements in deep learning have led to the development of sophisticated methods such as skip-connected convolutional autoencoders. These models have achieved remarkable results, with up to 98% accuracy in defect detection and a false pass rate below 1.7% ([Kim, Ko, Choi, & Kim, 2021](#)). Such methods not only detect defects but also assist in repairing them by highlighting discrepancies between input and output data ([Khalilian, Hallaj, Balouchestani, Karshenas, & Mohammadi, 2020](#)). The integration of machine learning and artificial intelligence into PCB manufacturing has yielded substantial benefits, notably in cost reduction and manufacturing efficiency. These technologies have significantly accelerated defect detection, minimizing waste and enhancing the reliability of electronic products. However, the integration of these advanced technologies into existing systems has introduced challenges, primarily due to high computational demands and the extensive data required for training. To mitigate these issues, various solutions have been implemented, including the optimization of algorithms to improve both

speed and accuracy, as well as the adoption of transfer learning strategies to decrease dependency on large volumes of labeled training data ([Ling & Isa, 2023](#)).

Chapter 3

Automating Human Operator Label in PCB Production Inspection Using ML/DL Models: The Critical Role of Synthetic Data Volume on Model Performance

In the literature review (Chapter 2), we discussed the role of inspection stations in ensuring the reliability and functionality of the final products during PCB manufacturing. It is important to acknowledge that defects can arise at various stages of the production line, each process carrying its inherent risks. To elaborate, each change made to an electronic item is followed by an inspection step such as Automated or manual Inspection to prevent defective boards from moving to the next step in the process.

In this chapter, we explore the manual inspection stage where human operators play a critical role in ensuring the quality of PCBs. Our focus is on predicting the human label generated using state-of-the-art machine learning and deep learning methods during the AOI stage of PCB production. By accurately predicting these labels, we aim to automate the manual inspection process,

thereby enhancing the efficiency and accuracy of defect detection in PCB production.

The subsequent sections of this chapter are structured as follows: Section 3.1 addresses the problem statement, while Section 3.2 provides an overview of the dataset employed in developing the proposed models in this chapter. Section 3.3 discusses the various steps taken in data preparation, such as data aggregation, cleaning, encoding, and feature selection. Additionally, the methods used for handling imbalanced data and their implementation are completely explained. Section 3.4 introduces the ML and DL models proposed for defect detection, along with a discussion of the implementation process. The evaluation of the model's effectiveness is presented in Section 3.5. Finally, Section 3.6 presents a summary and the concluding remarks for this chapter.

3.1 Problem Statement

As previously discussed, The PCB assembly journey is a complex, multi-stage operation that begins with printing the Panel Identification Number (PanelID) on each board. Following this initial step, a precisely measured amount of solder paste is applied to designated areas known as solder pads. This step is critical and demands meticulous accuracy; any deviation, such as applying an incorrect amount of solder paste or misalignment can compromise the functionality of the entire PCB, as depicted in Figure 3.1. To this end, upon completion of the solder paste application, the Printed Circuit Board undergoes Solder Paste Inspection to verify and ensure the quality of the soldering process. During the solder paste application, sensors collect data on various physical characteristics of the applied paste, which is then organized into tabular data displaying the quantity and precise placement of the solder paste on the pads.

While the AOI stage uses advanced imaging techniques to identify potential defects, it is not infallible. Hence, following this step, human operators manually examine PCB components to validate the AOI-generated labels and assign a new label referred to as "OperatorLabel." This manual step is essential to mitigate the risks of incorrect labeling, leading to significant costs if a good PCB is erroneously flagged as defective or if a faulty PCB progresses further down the production line. Given the critical nature of this step, we aim to automate this process by predicting the "OperatorLabel" in our dataset. This target variable has two possible values: "Good" and "Bad." The "Good"

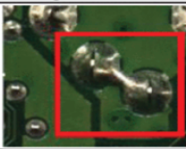

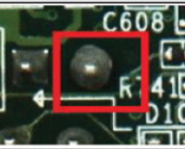

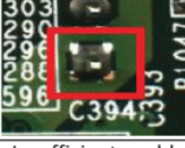

Defect Type	Bridge	Empty	Excess Solder
Sample			
Description	The solders are melted together.	Insufficient solder on the solder point.	Overflowed solder on the solder point.
Defect Type	Appearance_Hole	Appearance_Less	Appearance
Sample			
Description	Voids on pad region.	Insufficient solder on the solder point.	Dust or other material on the PCB board.

Figure 3.1: Common Defect Types of the PCB Board (Li et al., 2020)

label indicates that the AOI mistakenly detected a defect, and the component should not have proceeded to the inspection stage. Conversely, the "Bad" label confirms a genuine defect identified by AOI.

Although human inspection offers unique benefits in defect detection and operators can differentiate between false alarms and actual defects, leveraging ML and DL models provides significant advantages. These techniques can rapidly and consistently analyze vast amounts of data, reducing the likelihood of human error. These models can learn from patterns and anomalies, potentially identifying defects that might be missed by human operators, while maintaining continuous monitoring. This combination of speed, consistency, and the ability to detect minor variations makes ML and DL models invaluable for efficient and accurate defect detection.

The objective of the study in this phase is to develop ML and DL models capable of predicting the OperatorLabels, thereby enhancing the defect detection process. Integrating human expertise with automated inspection methods can bridge the gap between manual assessments and AOI systems, improving overall manufacturing efficiency and ensuring a more reliable PCB production process. Ultimately, this chapter assesses the effectiveness of ML and DL models compared to human operators in identifying PCB defects. This comparison will determine the viability of substituting human inspectors with automated systems, potentially yielding a more cost-effective and time-efficient quality control process in PCB manufacturing.

3.2 Dataset Overview

As previously outlined in Figure 2.1, the PCB production line we are analyzing consists of five stages. However, our data collection for defect detection specifically targets the SPI and AOI stages. The SPI dataset contains information about the solder paste applied by the machine, aiming to identify potential issues such as excess or missing paste and short circuits. This dataset includes key identifiers and quantifies specific attributes of the solder paste application. In parallel, the AOI dataset contains similar identifiers to the SPI dataset and provides three types of defect labels, each corresponding to different kinds of defects detected during the inspection process.

In the subsequent sections of this chapter, we will be delving deeper into both the SPI and AOI datasets, detailing their respective features and the significance of each in the defect detection process. By integrating and analyzing these datasets, we aim to enhance our understanding of the defect detection mechanisms and improve the overall efficiency and accuracy of the PCB production line. This thorough examination will facilitate the development of robust models capable of predicting and addressing defects at various stages, thereby ensuring high-quality PCB manufacturing.

3.2.1 SPI Dataset

The SPI dataset serves as a valuable source of information that can be effectively utilized in quality control, process improvement, and defect detection. It is used to examine the quality of solder paste applications. This step is essential for ensuring that the appropriate amount of solder paste is correctly applied to specific locations, which is crucial for forming reliable solder joints. In our study, the SPI dataset comprises 21 columns including categorical and numerical, each capturing distinct attributes pertinent to the inspection and testing of PCBs. A detailed explanation of each feature is presented in Table 3.1.

As shown in the table, the dataset includes identifiers such as PanelID, FigureID, ComponentID, and PadID. The combination of PanelID and FigureID forms the BoardID, representing the PCB. This valuable information can be effectively utilized in quality control, process improvement, and defect detection.

Table 3.1: Detailed Feature Overview of the SPI Dataset

	Feature	Unit	Description	Type
1	PanelID	-	Denotes the specific panel.	Categorical
2	FigureID	-	Denotes the specific Figure.	Categorical
3	ComponentID	-	Denotes the specific Component.	Categorical
4	PinNumber	-	Component's associated pin number.	Numerical
5	PadID	-	Unique ID for pin's supporting pad.	Categorical
6	Date	MM/DD/YYYY	Shows the SPI operation date.	Numerical
7	Time	HH:MM:SS	Time of SPI operation in seconds.	Numerical
8	PosX	mm	X coordinate of pin from bottom left.	Numerical
9	PosY	mm	Y coordinate of pin from bottom left.	Numerical
10	PadType	-	Specifies the type of pad.	Categorical
11	Volume	%	Percentage of the paste volume.	Numerical
12	Height	um	Height of the paste in micrometers.	Numerical
13	Area	%	Percentage of the paste area.	Numerical
14	OffsetX	%	X-axis offset percentage.	Numerical
15	OffsetY	%	Y-axis offset percentage.	Numerical
16	SizeX	mm	Size of the paste in the X direction.	Numerical
17	SizeY	mm	Size of the paste in the Y direction.	Numerical
18	Volume	um ³	Paste volume in cubic micrometers.	Numerical
19	Area	um ²	Paste area in square micrometers.	Numerical
20	Shape	um	Shape of the paste.	Numerical
21	Result	-	Outcome of SPI inspection.	Categorical

3.2.2 AOI Dataset

The AOI dataset serves as an additional crucial source of information, enhancing the data provided by the SPI dataset. It plays a crucial role in process monitoring, yield analysis, and traceability,

ensuring that each PCB meets the required quality standards before proceeding to the next stage of production. Similar to the SPI dataset, the AOI dataset includes PanelID, FigureID, and ComponentID identifiers. These common identifiers, along with PinNumber, facilitate merging the two datasets and allow for a more comprehensive analysis. Additionally, the AOI dataset includes three unique labels assigned to PCBs during various manufacturing stages, offering valuable insights for defect identification and classification. The details of the AOI dataset and its features can be found in Table 3.2.

Table 3.2: Detailed Feature Overview of the AOI Dataset

	Feature	Description	Categories
1	PanelID	Denotes the specific panel.	-
2	FigureID	Denotes the specific Figure.	-
3	ComponentID	Denotes the specific Component.	-
4	PinNumber	Component's associated pin number.	-
5	MachineID	Denotes the machine performing the AOI operation.	1) A 2) B
6	AOILabel	The label applied by the AOI machine based on the type of the defect.	1) Broken 2) Coplanarity 3) Jumper 4) LeanSoldering 5) Misaligned 6) Soldered 7) Translated 8) UnSoldered
7	OperatorLabel	The label applied by the human operator after visual inspection.	1) Good 2) Bad
8	RepairLabel	The label applied by the repairment operator after an inspection with a microscope.	1) Not Available 2) NotYetClassified 3) NotPossibleToRepair 4) FalseScrap

As indicated in the Table, machines, as optical inspection systems assess each PCB. They categorize the boards by assigning one of eight specific labels, known as **AOI labels**, based on how well each board adheres to predefined standards. Further details on AOI labels are elaborated below:

- **Broken:** This label is used when a pin has suffered physical damage or is completely detached, making it incapable of forming any electrical connections.
- **Coplanarity:** When pins are not level with one another, this can lead to inconsistent and unreliable electrical connections.
- **Jumper:** This refers to an accidental link between two pins, usually caused by solder bridging, which might lead to electrical shorts.
- **Lean Soldering:** Indicates insufficient solder or a suboptimal soldering process, resulting in a fragile mechanical or electrical bond.
- **Misaligned:** This label is assigned when a pin does not properly line up with its designated footprint or pad on the PCB, complicating or preventing a stable connection.
- **Soldered:** Normally indicates that a pin is correctly soldered; however, if marked as a defect, it suggests the quality of soldering is inadequate.
- **Translated:** Used when a pin has moved from its original position, either sideways or vertically, potentially compromising the connection's integrity.
- **UnSoldered:** This label is given when a pin has not been soldered, leaving it disconnected both electrically and mechanically from the board.

Given that there is always a potential for errors at each stage and results are not definitive, products proceed to the next stage which is **operator inspection** for further and more detailed inspection. In this stage, a specialist is responsible for re-evaluating the products. Ultimately, two labels—either “*Good*” or “*Bad*”—are assigned based on the product’s condition. Products labeled as “*Good*” when the AOI machine mistakenly marks them as defective, though it is in good condition and does not need any repairs, are removed from the process, while those labeled as “*Bad*” move on

to the **repair stage**. Following precise assessments by the specialist in the repair stage, each PCB is assigned one of four specific labels that describe its condition and dictate the next steps in the repair process. These labels are crucial for streamlining the repair operations and ensuring each board is treated appropriately. Below is a detailed explanation of each repair label:

- **Not Available (NA):** This label is applied to components that were initially misidentified as defective by the AOI machine but are later confirmed as compliant with quality standards by the Operator Label. As a result, these components are exempt from further repair assessments and are directly assigned the 'NA' label, indicating no further action is required.
- **Not Yet Classified:** Assigned to components for which repair data is still pending, this label signifies that the components have not yet been evaluated for repairability. It serves as a placeholder until further classification can be determined based on subsequent inspections or additional information.
- **Not Possible to Repair:** This label is used when it is determined that the damage to a pin is irreparable, leading to a recommendation to discard the entire panel. It highlights components that are beyond economic repair, underscoring the necessity of quality control in minimizing production losses.
- **False Scrap:** This label is designated to components that were originally flagged for defects by the operator; however, further evaluations indicate that no repair is necessary. It is a critical label that helps to reduce unnecessary waste and optimize resource use by ensuring that only genuinely defective components are reprocessed or discarded.

The data analyzed in this chapter consists of a combination of SPI and AOI datasets. Merging SPI and AOI data offers an integrated view of PCB manufacturing, highlighting how solder paste application impacts solder joint quality. By exploring the linkage between SPI metrics and AOI-detected defects, manufacturers can refine processes to decrease defect rates. Utilizing historical data supports predictive strategies to prevent potential production issues, enhancing both product quality and manufacturing efficiency.

3.2.3 Descriptive Analysis of The SPI-AOI Dataset

1. Statistical Summary of Feature Variables: Table 3.3 provides a comprehensive analysis of the employed dataset. As can be observed, the feature Volume(%) exhibits significant variability, with a standard deviation of almost 14% against a mean of 80%. This indicates that volume application is not uniform across different solder pads, potentially affecting the consistency of the final products. Conversely, Area(%) displays a tighter distribution around its mean, with a standard deviation of only 4.8 %, suggesting more consistent area coverage across the production line.

Critical insights arise from OffsetX(%) and OffsetY(%) features. The stretching ranges of these features from negative to positive indicate significant misalignments and outliers in solder paste placement. These deviations not only highlight the need for precise alignment in the manufacturing process but also pinpoint specific areas where process controls can be tightened to enhance product quality. Furthermore, the skewness in the distribution of Volume(um3) and Area(um2) with outlier values, suggests that while the majority of components meet standard specifications, a few of them exhibit exceptional variations. These anomalies could stem from unique manufacturing conditions or measurement errors, underscoring the importance of robust quality control measures. Such findings warrant further investigation to ensure these extremes do not compromise the overall production efficacy.

Table 3.3: Statistical Summary of Features

	Volume(%)	Height(um)	Area(%)	OffsetX(%)	OffsetY(%)	SizeX
mean	80.054	106.23	99.43	0.56	0.27	3.04
std	14.53	10.71	4.87	1.70	4.91	2.07
min	0.00	35.00	0.00	-16.70	-48.83	0.22
median	76.91	104.71	99.95	0.38	-0.14	3.30
max	289.22	313.99	166.01	37.52	52.36	5.10
	SizeY	Volume(um3)	Area(um2)	Shape(um)	PosX(mm)	PosY(mm)
mean	2.89	1323132	1299217	23.90	126.47	63.03
std	2.14	1254871	1230087	24.95	67.29	33.32
min	0.22	0	0	-35.00	9.90	14.70
median	2.80	5276616	4838858	28.50	132.50	55.10
max	5.10	7714679	22720260	332.00	236.10	125.80

2. Correlation Matrix: To enhance the understanding of the correlation among variables and potential predictive powers, an analysis of the correlation matrix was performed using the Pearson correlation coefficient. This statistical measure, which varies between -1 and 1, quantifies the degree of linear correlation between pairs of continuous variables. Coefficients approaching 1 signify a strong positive correlation, while those approaching -1 denote a strong negative correlation. Coefficients near zero imply an absence of linear correlation. As revealed in Figure 3.2, there

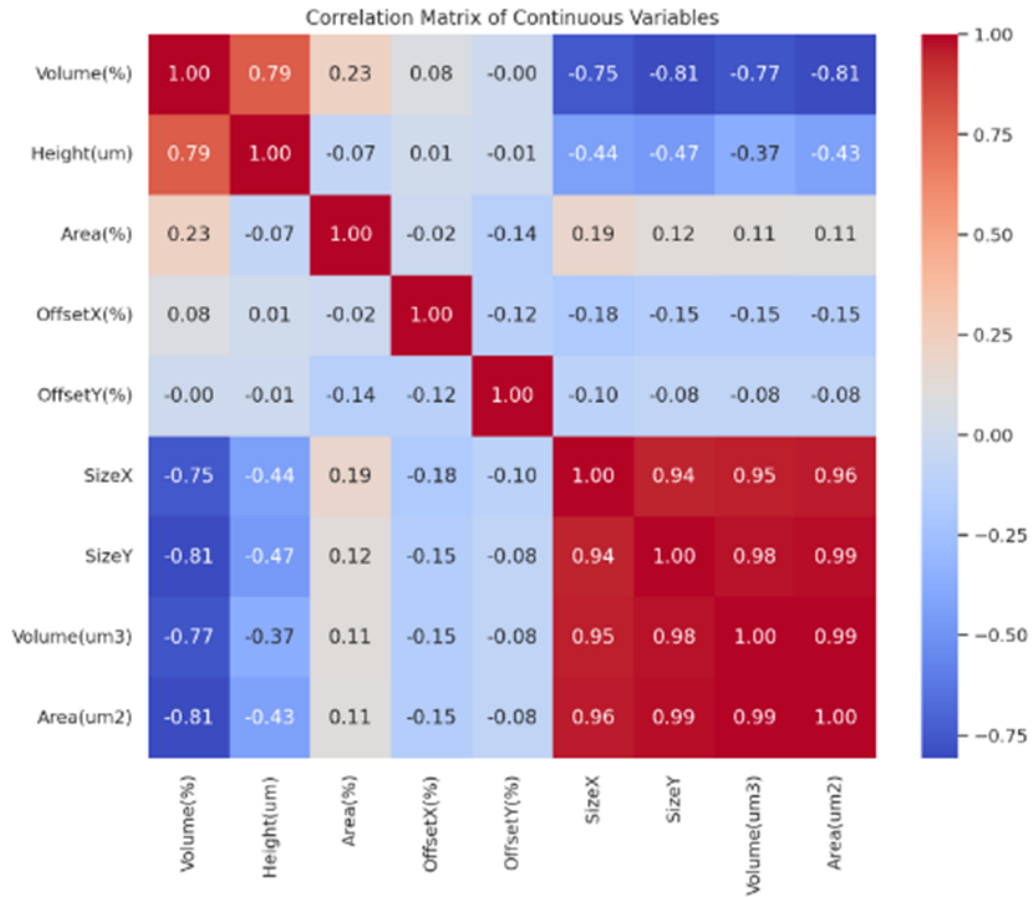


Figure 3.2: Continuous features correlation matrix

is a significant correlation among the attributes such as "sizeX," "sizeY," "Volume," and "Area", likely due to geometric dependencies. These variables displayed exceptionally high correlation coefficients, underscoring a robust linear relationship driven by the geometrical properties of PCB components. SizeX and SizeY, fundamental to the layout and spatial requirements of components, directly influence the calculated Volume and Area. Volume, is conceptualized as the product of

SizeX, SizeY, and height. Similarly, Area derived from SizeX multiplied by SizeY, shows nearly perfect correlations due to these geometric dependencies.

Interestingly, despite the strong correlations among Size and Volume metrics, OffsetX(%) and OffsetY(%) exhibit low correlations with these dimensions, suggesting that positional deviations are influenced by factors unrelated to component size, such as placement machinery precision or variability in component handling.

This nuanced understanding of feature correlations is pivotal for refining predictive models and optimizing production algorithms, ensuring they are both effective and computationally efficient. The detailed correlation study not only reinforces the geometrically proportional nature of the variables but also provides crucial insights into the physical structure of the dataset and the real-world processes it represents, thereby enhancing our capability to optimize production parameters for improved consistency and reliability in PCB assemblies.

3.2.4 Categorical Variables Exploration

Following the analysis of continuous variables, this section offers an in-depth examination of the distribution of categorical features within the SPI-AOI dataset. This analysis concentrates on three pivotal categorical columns—AOILabel, Result, and MachineID—along with essential target labels, including OperatorLabel and RepairLabel. Visual depictions clarify the proportional distribution of each category, exposing inherent patterns and potential biases within the data. Such an analysis is crucial as it highlights categorical features that provide substantial insights into outcomes, subsequently guiding our predictive modeling techniques and operational strategies.

As discussed earlier in Section 3.2, the AOI dataset comprises various unique labels that denote different types of defects. The AOILabel and Result distributions, as showcased in Figure 3.3, are characterized by high cardinality, posing challenges for modeling. The AOILabel pie chart reveals that a large portion of the PCB components are labeled as 'Soldered', while notable quantities are classified as 'UnSoldered' and 'Lean Soldering'. The infrequency of categories such as 'Jumper', representing only 0.2% of the data, underscores the issue of high cardinality, where many categories contain very few samples. Similarly, the Result distribution predominantly features defect-free components, with other defect categories like 'W.InSuffi.' and 'EPosition' appearing less frequently.

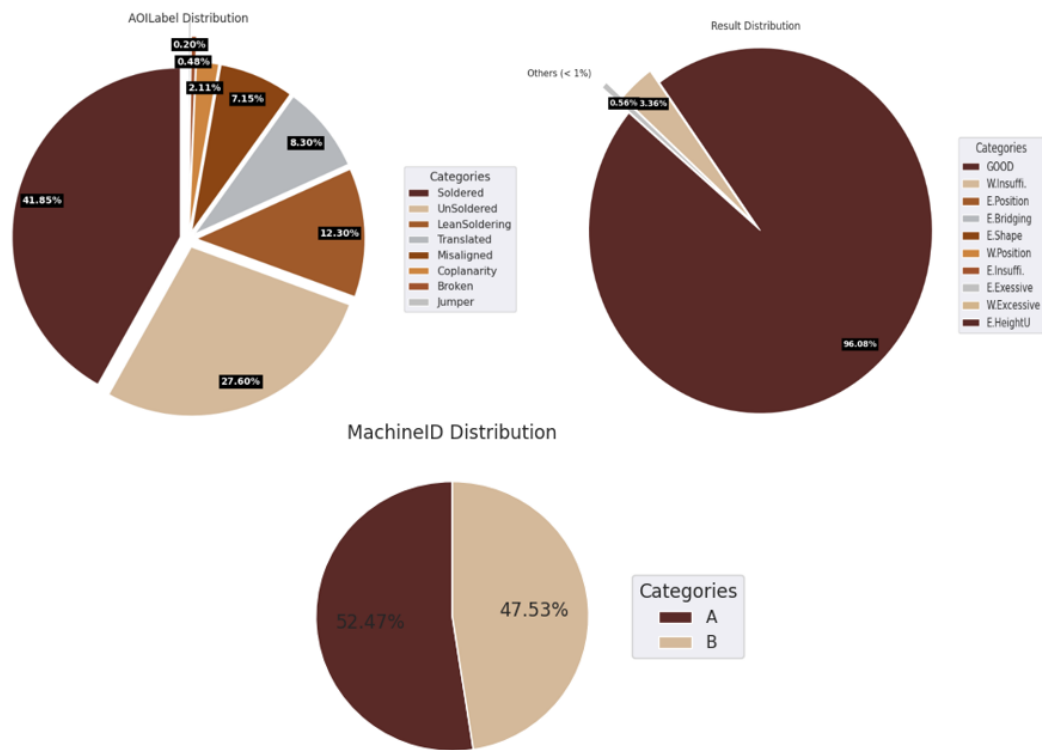


Figure 3.3: Categorical Features Labels

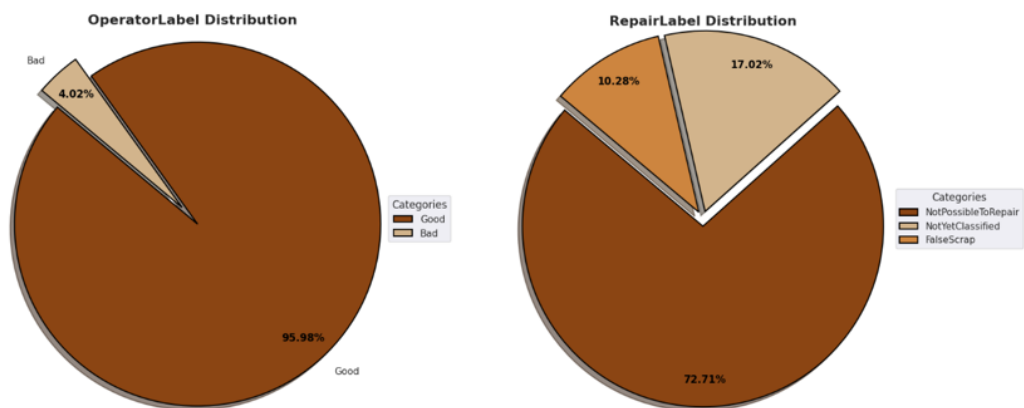


Figure 3.4: Target Features Distribution

On the other hand, the examination of target variables, as shown in Figure 3.4, reveals significant class imbalances critical for steering the predictive modeling process. The OperatorLabel distribution is highly skewed, with nearly 96% of components categorized as 'Good' and only about 4% as 'Bad'. Furthermore, the RepairLabel distribution indicates that the majority of components are categorized as 'NotPossibleToRepair', suggesting that a significant portion of defects are considered irreparable. The categories 'NotYetClassified' and 'FalseScrap' account for approximately 17% and 10% of the data, respectively, highlighting less frequent outcomes. This skewed distribution calls for sophisticated modeling strategies capable of effectively detecting the rarer defective cases, to improve the accuracy and robustness of predictions related to the outcomes.

3.3 Data Preparation and Pre-processing

This section meticulously outlines the essential steps required to prepare our datasets for comprehensive machine learning and deep learning analysis. We detail a series of crucial data preparation and preprocessing techniques, including data aggregation to prepare unified data to use, data cleaning to ensure the accuracy and consistency of our datasets, data augmentation for handling imbalanced data, encoding categorical data to make it suitable for analysis, and scaling features to standardize the range of independent variables. Additionally, we emphasize the importance of feature selection to identify the most significant variables and ensure that our models can learn effectively from the data. These preparatory measures are foundational, setting the stage for proposed models to perform precise, robust, and efficient analyses, ultimately enhancing our research outcomes' reliability and predictive power.

3.3.1 Data Aggregation

Since the SPI dataset is massive, it has been divided into four CSV files associating train sets and two test sets. Hence, as an initial step for making a singular training and testing SPI dataset, we began by combining the CSV files. After merging these files, the SPI dataset available for our study includes 21 features, split into almost 6,000,000 data points for training and 2,500,000 for testing. The AOI dataset, on the other hand, is relatively smaller and consists of one CSV file each for the

train and test sets, comprising eight features with approximately 31,600 entries for training and 13,600 for testing.

As noted earlier in Section 3.1, our primary objective at this stage is the accurate prediction of operator labels. Given that the AOI dataset primarily contains labels without additional descriptive features, it is necessary to combine it with the SPI dataset, which is rich in features. To this end, a crucial merging of the SPI and AOI datasets was executed during the data preparation phase to create unified datasets for both training and testing purposes. In this process, identifiers such as PanelID, FigureID, ComponentID, and PinNumber were concatenated to form a unique identifier for each entry across both datasets. This unique identifier served as a common key, facilitating the dataset's merging by aligning them based on these four shared attributes. Following the merging, non-informative columns such as Date, Time, and ConcatenatedID were removed from the dataset to streamline the data further. Additionally, the 'Repair Label', which represents the final inspection stage, was identified as irrelevant for this analysis phase and consequently excluded from the dataset. This strategic exclusion allowed us to focus solely on variables that directly impact the primary objective of predicting the 'Operator Label'.

Noteworthy, it is crucial to consider the role of identifier features such as PanelID, PadID, and FigureID carefully. Due to their high cardinality, PanelID and PadID were excluded from the dataset as their inclusion would substantially increase its dimensionality, complicating the analysis within an already extensive dataset. Preliminary evaluations revealed that PanelID lacked significant predictive power or relevance to the target variable, potentially introducing noise and misleading the model training process. Consequently, the exclusion of these identifiers mitigates the risk of overfitting and simplifies the interpretability of the results, allowing the models to concentrate on features that more directly affect the quality and characteristics of the PCBs. Conversely, FigureID was retained as a primary key because of its perceived relevance in analyzing component-specific trends and quality assessments within the dataset. To empirically validate this decision, we conducted additional experiments where FigureID was removed from the dataset. This resulted in a noticeable reduction in model performance, confirming the feature's predictive importance. Its inclusion is intended to enhance the model's ability to identify subtle patterns and variations among different PCB components, which is crucial for effective defect detection. This selective retention underscores the

strategic nature of our data preprocessing efforts, ensuring that each step is aligned with the overarching objectives of the study and tailored to the specific nuances of our dataset. Each preprocessing decision, including the removal and retention of specific features, was validated through rigorous testing to observe their impact on the predictive performance of our models, further grounding our approach in empirical evidence.

This refinement resulted in a consolidated dataset comprising 20 distinct features across 22,532 records. By integrating critical features from both inspection processes, this dataset provides a robust foundation for subsequent machine learning and deep learning analyses aimed at detecting quality issues in PCB manufacturing.

3.3.2 Data Cleaning and Integrity Verification

During the data verification phase, our first action is to ensure the dataset is devoid of extraneous and duplicate records, and that all entries are complete. This initial scrutiny confirmed there were no missing values, thus eliminating the need for imputation. However, 43 duplicate records were discovered and promptly removed to maintain the dataset's uniqueness and integrity. The dataset indices were then refreshed, promoting a systematic organization that enhances the reliability of subsequent data manipulations. In addition, Further measures were implemented to ensure all features were appropriately prepared for effective data analysis. A key adjustment was made to the 'PinNumber' feature, which typically indicates the number of pins for each component type. Notably, PinNumbers labelled as "THERMAL1" during the soldering process represent an incompatible format with the numeric requirements of our machine learning models. To rectify this, we converted all "THERMAL1" entries into a numeric value of "0". This change not only preserved the unique status of these PinNumbers but also ensured numeric consistency throughout the dataset, further solidifying our data's foundation for accurate analysis.

3.3.3 Feature Consistency Adjustments

Critical to the robustness of the predictive models is the prevention of data leakage, which necessitates the separation of data into distinct training and testing sets with identical variables. This measure is fundamental to avoiding the inadvertent blending of training and testing data, which could

compromise model evaluations. Detailed examination identified inconsistencies in the 'Result' and 'ComponentID' columns between the training and testing datasets. To align these sets accurately, specific modifications were applied: entries such as 'E.Bridging', 'W.Position', 'W.Excessive', and 'E.Height' were removed from the "Result" column of the training set. Additionally, the 'ComponentID' column observed the removal of entries such as C12, C3, D8, DZ1, L3, R41, TR2, TRB2, and TRB7 in the training set, and C32, C4, and R8 in the testing set. These modifications ensure uniformity across both datasets, crucial for the seamless functioning of machine-learning models. The adjustments made for harmonizing the feature values resulted in the removal of only 0.3 percent of the training data and 0.08 percent of the test data. These percentages are considerably low and are deemed negligible, thus maintaining the substantial bulk of data for effective machine-learning applications.

3.3.4 Feature Scaling

Data standardization is a pivotal preprocessing step in machine learning. It ensures that features within a dataset are normalized to a uniform scale, significantly enhancing the accuracy and efficiency of state-of-the-art algorithms (Żbikowski & Antosiuk, 2021). This section explores several scaling methods—MaxAbs Scaling, Min-Max Scaling, Standard Scaling, and Robust Scaler—each tailored for different dataset characteristics and offering unique benefits.

- (1) **MaxAbs Scaling:** This method normalizes each feature by scaling the data to the range of $[-1, 1]$, as represented in Formula 1. MaxAbs Scaling is especially beneficial for algorithms like k-Nearest Neighbors (k-NN), where it enhances overall performance by maintaining the data's original layout and preserving any inherent sparsity (Ahsan, Mahmud, Saha, Gupta, & Siddique, 2021). This property makes it a suitable choice for sparse datasets.

$$X_{\text{scaled}} = \frac{X}{\max(|X|)} \quad (1)$$

where x is the original value and $\max(|X|)$ represents the maximum absolute value within the feature column.

- (2) **Min-Max Scaling:** Min-Max Scaling linearly adjusts the data, normalizing feature values

to a predetermined range, typically $[0, 1]$, as shown in Formula 2. This method is highly susceptible to outliers, which can compress the bulk of the data into a narrow range (Nkik-abahizi, Cheruiyot, & Kibe, 2022). It is important to carefully consider this characteristic when applying Min-Max Scaling, particularly in datasets with significant outliers.

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (2)$$

where X_{\min} and X_{\max} are respectively minimum and maximum values.

- (3) **Standard Scaling (Z-score Normalization):** Standard Scaling adjusts data by removing the mean and scaling to unit variance, as indicated in Formula 3. This scaling method transforms the data distribution into a zero mean and a variance of one (Cao, Stojkovic, & Obradovic, 2016). However, the presence of outliers can significantly distort the mean and standard deviation, rendering this method less effective for outlier-rich datasets.

$$X_{\text{scaled}} = \frac{X - \mu}{\sigma} \quad (3)$$

where x is the original value, μ is the mean and σ is the standard deviation.

- (4) **Robust Scaler:** The Robust Scaler modifies feature values by subtracting the median and dividing by the interquartile range (IQR), making it less susceptible to outliers, as shown in Formulas 4 and 5. This method is particularly advantageous for datasets with extreme values.

$$X_{\text{scaled}} = \frac{X - \text{median}}{\text{IQR}} \quad (4)$$

$$\text{IQR} = Q3 - Q1 \quad (5)$$

where median is the middle value, and IQR is the difference between the third and first quartiles.

Given the significant presence of outliers and the varied data distributions in our PCB dataset,

a careful selection of the scaling method is necessary. After reviewing different techniques and considering the specific properties of our data, the Robust Scaler was selected as the most suitable method. Its ability to diminish the impact of outliers by using the median and IQR ensures that it effectively normalizes data without distorting crucial outlier effects. This characteristic, coupled with its robustness, makes the Robust Scaler superior to Min-Max and Standard Scaling for our dataset, and even more suitable than MaxAbs Scaling, which, while preserving sparsity, does not specifically address outlier sensitivity. The implementation of the Robust Scaler in our preprocessing pipeline has significantly enhanced the performance of the machine learning models employed in this study. Such standardization proves particularly advantageous prior to the introduction of data into data augmentation, the subsequent phase of preprocessing. It facilitates improved model convergence and reduces the impact of outliers, thus optimizing the quality of the inputs fed into the generative model.

Furthermore, using CTGAN for generating fake data, with its proficiency in handling complex data structures, obviates the need for traditional data transformations such as one-hot encoding for categorical variables. Differing from conventional models, this method can inherently process categorical columns, such as 'ComponentID', 'Result', and 'AOILabel', by internally transforming them into numerical formats. This attribute highlights the critical need for ensuring the cleanliness, consistency, and correctness of categorical data before it is introduced into any modeling process. The integrity of these data types is paramount, as it significantly influences the quality of the synthetic data generation. By diligently preparing our data through scaling and ensuring robust handling of categorical data within CTGAN, we establish a strong foundation for the crucial subsequent data augmentation phase.

3.3.5 Data Augmentation

Imbalanced datasets, characterized by a predominant number of non-defective samples relative to defective ones, tend to bias classifiers towards predicting the more frequent class. This bias can compromise the detection of the minority class, which is often crucial for identifying defects ([Park, Kwon, & Jeong, 2023](#)). As illustrated in Figure 3.4, the distribution of classes within the "Operator Label" target variable in our dataset is highly uneven, with healthy samples far outnumbering

defective ones. Such imbalance heightens the risk of overfitting, whereby models excel on training data but falter on new, unseen datasets. To mitigate this, balancing the dataset is critical to bolster the model’s capacity to generalize across diverse testing environments.

To rectify this imbalance, various techniques can be employed, broadly classified into data-level interventions and more sophisticated methods like those involving GANs. Reflecting the specific characteristics of our dataset, we have opted for two distinct approaches including SMOTE and CTGAN. These strategies will be detailed in subsequent sections, emphasizing their importance in developing models that deliver consistent and reliable predictions across all classes, thus enhancing both the robustness and the credibility of the outcomes.

1. Synthetic Minority Oversampling (SMOTE): The Synthetic Minority Oversampling Technique is a pivotal method devised to tackle the issue of imbalanced datasets. Unlike conventional oversampling methods that merely replicate existing minority instances, SMOTE generates new synthetic samples by interpolating between existing instances. This approach fosters a more varied and generalized representation of the minority class, thus enhancing the performance of classification algorithms (S. Wang et al., 2023). This method operates by first selecting an instance x_i from the minority class and identifying its k nearest neighbors x_{zi} within the same class. For each selected instance x_i , synthetic samples are created by interpolating between x_i and one of its nearest neighbors x_{zi} . This is achieved using the following formula:

$$s = x_i + \lambda \cdot (x_{zi} - x_i) \quad (6)$$

where λ is a random value between 0 and 1. This ensures the synthetic samples are positioned along the line segment connecting x_i and x_{zi} . Figure 3.5 visually illustrates this process, showing how synthetic samples are generated between existing minority class instances. Using This technique, we investigated the impact of the amount of synthetic data on the ultimate performance of the modeling process. The results are documented in Section 3.5.2, illustrating how varying levels of synthetic data influence model outcomes.

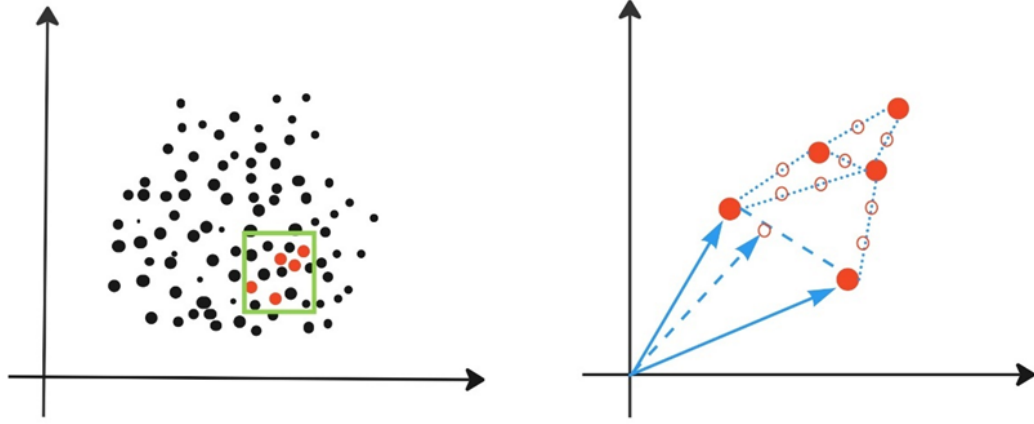


Figure 3.5: SMOTE Oversampling Process

2. Conditional Tabular Generative Adversarial Network (CTGAN): Although traditional methods such as SMOTE have effectively balanced datasets and enhanced model performance, recent advances in GANs offer new possibilities for improving PCB defect detection through the creation of realistic synthetic defect images. These innovative approaches have proven to be effective solutions for addressing data imbalance challenges within machine learning contexts. GANs are capable of generating high-quality synthetic data, often outperforming conventional oversampling techniques. However, the deployment of GANs involves certain challenges; issues like model collapse (Yeom, Gu, & Lee, 2024), poor quality (Laria, Wang, van de Weijer, & Raducanu, 2022; Ye, Wang, & Chen, 2023), and intensive computational resources (Verma, Arora, & Perumal, 2023) represent significant obstacles. Despite these challenges, the latest developments in GAN technology have demonstrated considerable promise in enhancing the detection of defects in PCB manufacturing. In response to these developments, this part is dedicated to a thorough evaluation and exploration of CTGAN application, assessing their ability to address the problems associated with imbalanced datasets.

In this section, we will initially provide an overview of how the CTGAN model functions, establishing a foundational understanding of its architecture. Subsequent to this theoretical introduction, we will detail the implementation of the model, discussing how it was applied to our dataset. The discussion will include the specific libraries utilized, the analytical methods employed, including

illustrative charts, and a comprehensive evaluation of the generated data's validity. This detailed presentation is designed to offer a clear insight into the model's effectiveness and its practical utility in addressing real-world data challenges.

(1) CTGAN Architecture

GAN architecture, as illustrated in Figure 3.6, is composed of two key deep neural network components: the generator (G) and the discriminator (D). The generator functions as a generative model that produces data from a noise input, while the discriminator acts as a classification model that determines whether the input data is real or generated (Mendikowski & Hartwig, 2022). Traditional GANs often encounter difficulties when dealing with diverse data types, non-Gaussian distributions, multimodal data, sparse matrices resulting from one-hot encoding, and categorical variables with significant imbalances. To overcome these challenges, the CTGAN model was developed, which harnesses the power of GANs specifically for structured data environments.

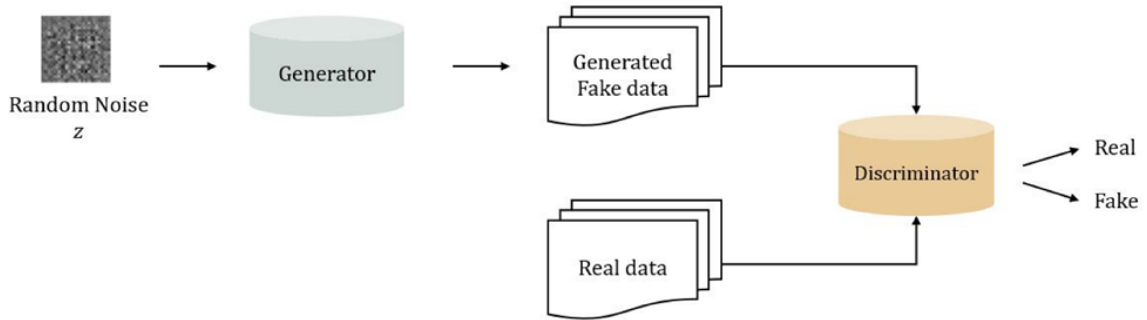


Figure 3.6: Structure of GAN (Eom & Byeon, 2023)

This model combines elements from both conditional GANs and tabular GANs to form a powerful model designed to manage the complexities of tabular data. This is achieved through two primary techniques: mode-specific normalization and training-by-sampling.

- **Mode-specific normalization:** This technique addresses challenges associated with non-Gaussian and multimodal distributions in numerical data by applying a variational Gaussian mixture model (VGM) to normalize the data. This normalization ensures that the input provided to the generator accurately reflects the statistical properties of the dataset, making

it more suitable for the generative process.

- **Training-by-sampling:** This technique tackles issues related to categorical features by ensuring a balanced representation of categories. It modifies the sampling process according to the logarithmic frequency of the categories, which allows the model to thoroughly explore and learn from all possible discrete values in the dataset. The specific process is illustrated in Figure 3.7. By employing these approaches, CTGAN can generate synthetic data that is diverse and representative of the original dataset, thereby improving model training and overall performance.

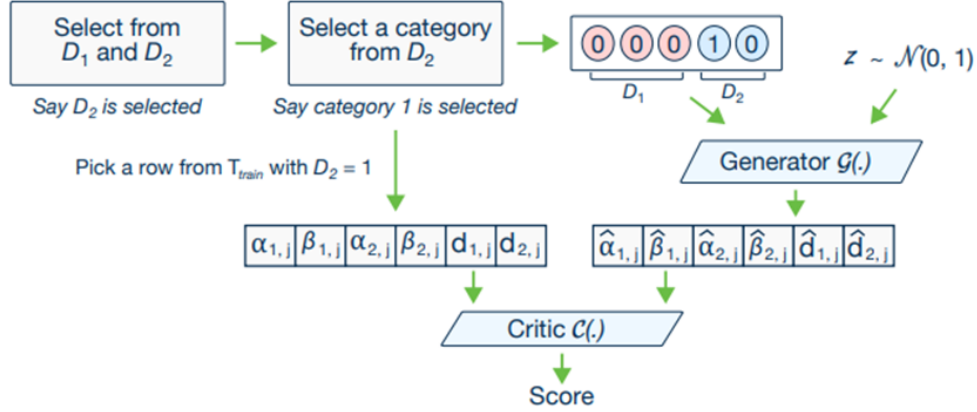


Figure 3.7: Training-by-sampling of CTGAN (L. Xu et al., 2019)

Given the specific characteristics of our dataset, which includes both categorical and continuous features, non-Gaussian distributions, multimodality, and a high level of imbalance, the advantages of this model make it an ideal choice for our needs. CTGAN’s ability to effectively model complex data distributions and generate high-quality synthetic data while handling the inherent challenges of our dataset is why we selected it for this study. Following this rationale, we now proceed to the implementation phase.

(2) CTGAN Implementation

In the implementing phase, we leveraged the Synthetic Data Vault (SDV) library (Patki, Wedge, & Veeramachaneni, 2016), which provides a robust framework for generating synthetic data through various generative models, including CTGAN. SDV library is particularly

suit for handling complex data structures and distributions, making it an excellent choice for our dataset's specific challenges. We also incorporated the 'SingleTableMetadata' to efficiently manage dataset metadata, which is crucial for the model's understanding of different column types and roles within the data. Upon loading the dataset, metadata was automatically extracted from the data frame, converted into a dictionary format for easier inspection, and then validated against the actual data to ensure accuracy and consistency, thus preventing discrepancies during the data generation.

In the training architecture of the model, two fully connected hidden layers are utilized for both the generator and discriminator components, enhancing the model's ability to learn and synthesize complex data patterns effectively. Notably, the activation functions play a pivotal role in regulating the flow of gradients through the network, thus aiding the optimization process. In the generator, **ReLU (Rectified Linear Unit)** activation function is employed for its simplicity and efficiency, which facilitates a faster and more effective training process by allowing only positive values to pass through, thus mitigating the vanishing gradient problem. Additionally, **Leaky ReLU** is implemented in discriminator to prevent the 'dying ReLU' problem by allowing a small, non-zero gradient when the unit is inactive, ensuring that all neurons remain functional and contribute to the learning process. Conditional sampling was strategically employed within the CTGAN framework, ensuring the model could produce data under predefined conditions. This was particularly crucial for aligning the number of defective labels with the normal ones in our preprocessed dataset which comprised 21,551 normal labels and only 858 defective labels. To rectify this imbalance and enhance the model's predictive accuracy, we generated an additional 20,693 defective labels with 20 features, thus ensuring a balanced training environment. To optimize the hyperparameters and ensure it is precisely adapted to the specific characteristics of our dataset, we utilized **GridSearch**. This method involved testing a range of values for critical parameters such as learning rate, batch size, and the number of training epochs. The ranges for these parameters were determined based on preliminary tests and insights drawn from the literature. Ultimately, the best-identified hyperparameters are 'embedding_dim': 128, 'discriminator_steps': 2, 'epochs': 594 with 'batch_size': 500. To further assess the effectiveness of the training

process, a detailed loss analysis for both the generator and discriminator will be presented in the subsequent part.

- **Training Dynamics and Loss Analysis:** Fundamentally, the dynamics between the generator and discriminator in CTGAN training reflect the continuous adversarial interactions described by (Liu & Hsieh, 2019). The efforts of the generator to produce increasingly realistic data and the discriminator’s role in differentiating real from synthetic data lead to the observed fluctuating loss patterns. These fluctuations, common in GAN training as noted by (Mescheder, Geiger, & Nowozin, 2018), indicate ongoing adjustments and learning within the model rather than failures. Further, the stabilization methods applied by (K. Xu, Li, Zhu, & Zhang, 2019) using control theory underscore the intrinsic nature of these fluctuations as part of the iterative learning process between G and D.

Building on these insights, Figure 3.8 in our study provides a detailed analysis of the CTGAN training dynamics, showcasing the significant impact of varying hyperparameter configurations on the model’s operational efficacy. This figure includes a series of loss graphs for different settings, vividly illustrating typical patterns of loss fluctuations and their convergence. Notably, the configuration resulting in the most stable and effective training outcomes demonstrates an ideal balance between the losses of the generator and discriminator, smoothly converging to a consistent mean. Initially, significant fluctuations stabilize as training progresses. This stabilization around a consistent mean indicates effective mutual adaptation and learning between the components, essential for achieving a Nash equilibrium.

Furthermore, the loss oscillations remain confined within a predefined range, showing no signs of extreme volatility or divergence, thereby avoiding complications such as mode collapse—where the generator might fail to produce more than a limited array of outputs. This stability suggests that the chosen hyperparameters are finely tuned to foster an environment conducive to effective training. The discriminator updates at a carefully moderated pace, enhancing the generator’s ability to produce diverse and realistic data, crucial for generating

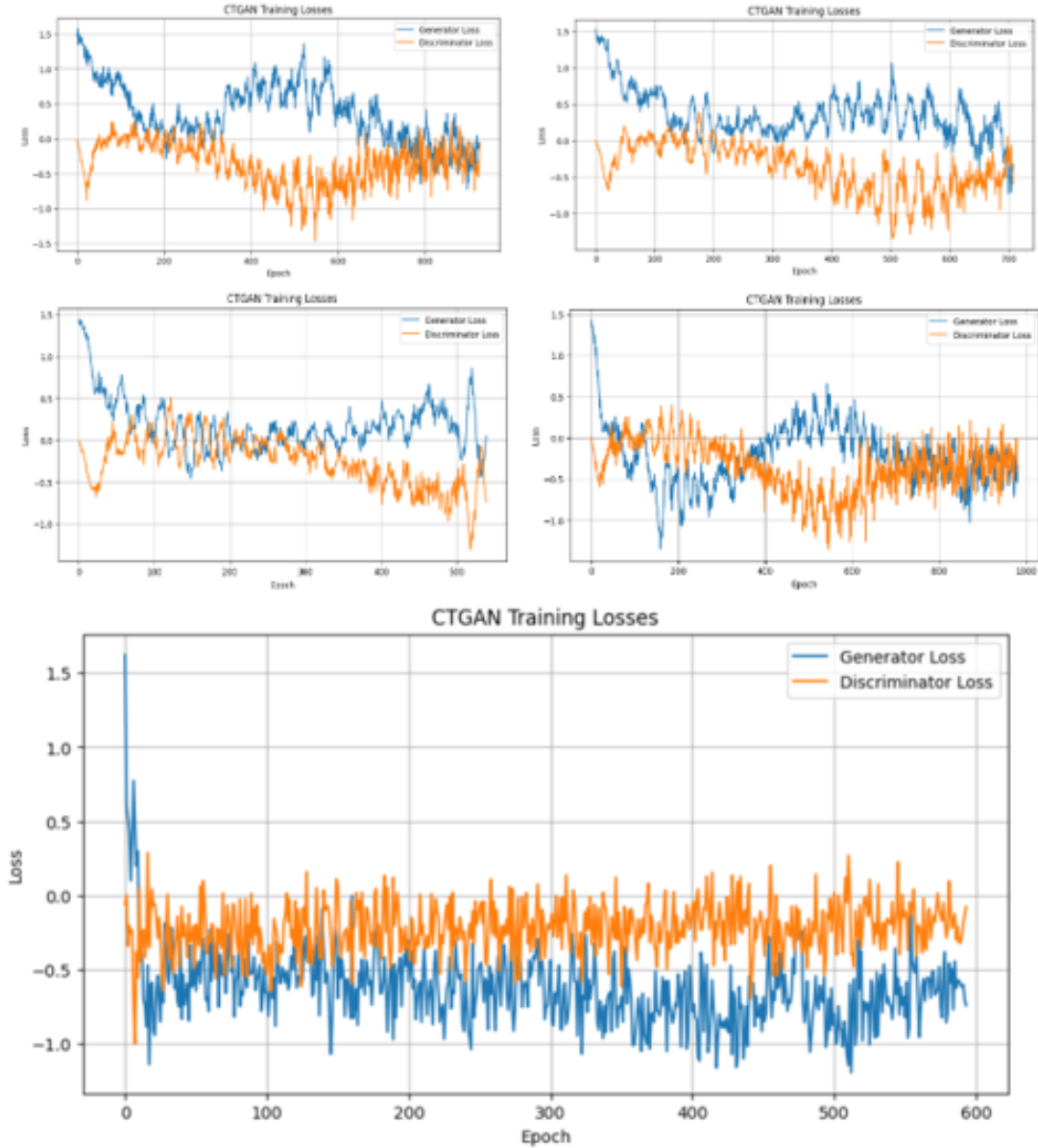


Figure 3.8: Loss values of CTGAN training process

high-quality synthetic data. These empirical findings underscore the importance of strategic hyperparameter tuning in optimizing model performance and output quality, confirming the critical role of fine-tuning hyperparameters to achieve stable training outcomes.

(3) CTGAN Synthetic Data Evaluation

After generating synthetic data, it is essential to evaluate how closely this data replicates the real dataset's statistical properties, distributions, and correlations. This evaluation step is crucial for ensuring that the synthetic data can be reliably used for further analysis, testing, or training machine learning models without introducing biases or significant deviations from real-world data.

- **Diagnostic Report:** The diagnostic validation process was conducted with meticulous attention to the validity and structural integrity of the data, including stringent checks to ensure that continuous columns conformed to the minimum and maximum values of the real data. Similarly, categorical columns were verified to contain the same categories as those present in the actual dataset. This thorough examination resulted in an overall score of 100%, confirming that the synthetic data adheres flawlessly to the real data and precisely replicates its structural intricacies.
- **Quality Report:** As Table 3.4 illustrates, the quality report for the synthetic data generated yielded an overall score of almost 84%, reflecting a strong replication of the original dataset's characteristics while also highlighting areas for improvement. The evaluation measures the statistical similarity and focuses on two key aspects: Column Shapes and Column Pair Trends. The Column Shapes score close to 88%, indicates that the synthetic data effectively captures the univariate distributions and statistical properties of most columns in the original dataset. However, the Column Pair Trends score of 80 %, reveals a few discrepancies in accurately modeling the bivariate interactions and correlations between columns. This suggests that while the synthetic data is reliable for analyses and applications that are less sensitive to inter-variable dependencies, further refinement may be necessary to enhance the accuracy of these complex relationships for tasks that depend heavily on such correlations. Improving these aspects could involve refining the CTGAN model's training process or experimenting with different configurations to better capture the nuanced dynamics of the dataset.

To better understand the discussed results and visual comparison, the distribution plots for some selected continuous and categorical variables are provided in Figures 3.9 and 3.10.

Table 3.4: Synthetic Data Quality Report

Column Shape	Column Pair Trends	Overall Quality Score
87.74	80.01	83.87

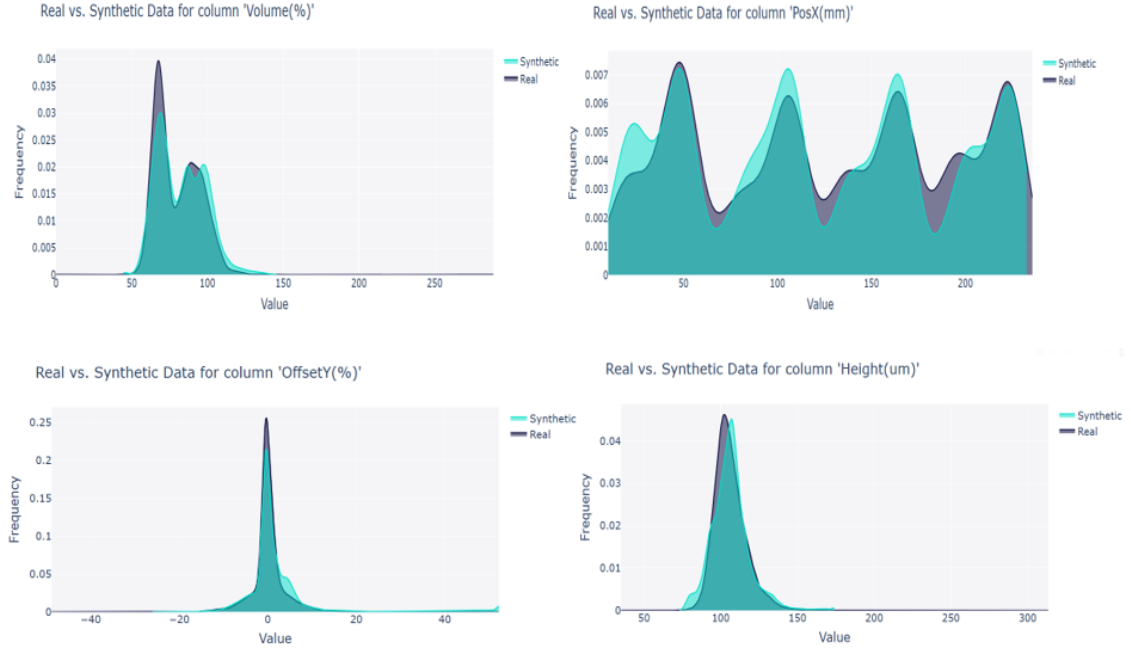


Figure 3.9: Continious Features Comparison of the generated data and actual data

As can be observed in the above histograms, the distributions in the synthetic data closely match those of the real data, which is a good indicator. The key is that even without complete loss convergence, the CTGAN model generated high-quality synthetic data, capturing the underlying distributions of the real dataset. Further analysis was conducted using Total Variation Distance and Kolmogorov-Smirnov complement metrics to evaluate the quality of augmented data for categorical and numerical features, respectively.

- **Total Variation Distance Metric:** Total Variation Distance (TVD) is a robust metric extensively utilized for assessing the congruence between the distributions of synthetic categorical data and their real counterparts (Corander, Remes, & Koski, 2021). This metric is widely applied across different fields to ensure that synthetic data accurately mirrors the distributional characteristics of real data. For instance, (Regol, Kroon, & Coates, 2023)

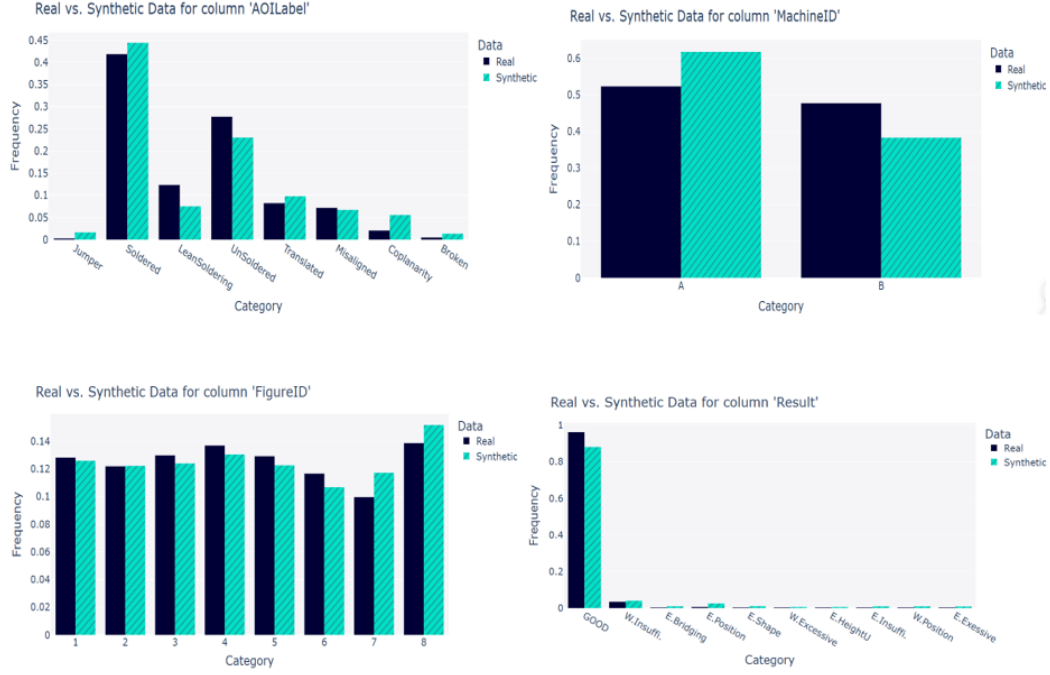


Figure 3.10: Categorical Features Comparison of the generated data and actual data

and (Ackerman, Kour, & Farchi, 2023) have explored the application of this metric in assessing generative models that produce categorical data. They emphasize the significance of TVD in diminishing the disparities between the distributions of real and synthetic data across various use cases. The formula for calculating TVD between two discrete probability distributions, P and Q , over a discrete variable X is expressed as follows:

$$\text{TVD}(P, Q) = \frac{1}{2} \sum_{x \in X}^n |P(x) - Q(x)| \quad (7)$$

- **Kolmogorov-Smirnov Metric:** The Kolmogorov-Smirnov (KS) test is a non-parametric statistical technique designed to assess the degree of similarity either between a sample and a reference distribution or between two independent samples. It quantifies this similarity by measuring the maximum deviation between their empirical cumulative distribution functions (ECDFs). The KS test is especially valuable for verifying how well synthetic data replicate the distribution of real data. Higher KS values suggest a closer match between the distributions (Bai & Kalaj, 2021). The formula for the KS statistic, known as D statistic, is

given by:

$$D_n = \sup_x |F_n(x) - F(x)| \quad (8)$$

where $F_n(x)$ is the ECDF of the sample, $F(x)$ is the CDF of the reference distribution and \sup denotes the maximum value of the set of distances $|F_n(x) - F(x)|$.

Table 3.5 outlines the scores for the KS test and TVD metrics across various attributes of the datasets. Higher scores suggest a greater congruence between the synthetic and real data distributions. Notably, the attribute “FigureID” registers a high TVD score of 0.95, demonstrating a strong match. Similarly, “PinNumber” and “Volume(%)” attain high KS scores of 0.95 and 0.94, respectively, indicating a close resemblance between the synthetic and actual data distributions for these attributes. Conversely, attributes such as “SizeX” and “Shape(um)” display lower scores of 0.68 and 0.69, respectively, signalling potential areas for enhancement in the synthetic data generation process.

Overall, these metrics provide a quantitative analysis of how well the synthetic data replicates the real data across different features, highlighting areas where the synthetic data generation aligns closely with the real data and areas where improvements might be necessary. This analysis is crucial for validating the utility of synthetic data in applications where accurate reproduction of real-world data distributions is critical.

Table 3.5: KS and TVD Metrics and Scores

Metric	Feature Name	Score	Metric	Feature Name	Score
KS	PinNumber	0.95	KS	Shape(um)	0.69
KS	Volume(%)	0.94	KS	PosX(mm)	0.91
KS	Height(um)	0.92	KS	PosY(mm)	0.86
KS	Area(%)	0.90	TV	FigureID	0.95
KS	OffsetX(%)	0.91	TV	ComponentID	0.83
KS	OffsetY(%)	0.88	TV	PadType	0.90
KS	SizeX	0.68	TV	Result	0.89
KS	SizeY	0.73	TV	MachineID	0.90
KS	Volume(um3)	0.80	TV	AOILabel	0.90
KS	Area(um2)	0.86	TV	OperatorLabel	0.88

3.3.6 Data Encoding

Following the initial data preparation and data generation, transforming data into a format that is interpretable by ML models is critical. This is especially true for categorical variables, which inherently lack a numerical representation and thus pose a unique challenge for ML models. To address this, two predominant methods are employed: label encoding and one-hot encoding. Label encoding assigns a distinct integer to each unique category within a variable, effectively transforming categorical data into a numerical format. However, this method may inadvertently imply an ordinal relationship among the categories, which might not exist in reality. To prevent such potential misinterpretations, one-hot encoding is often utilized. One-hot encoding involves creating a separate binary column for each category, thus eliminating any implied order or hierarchy among them. By integrating both techniques, categorical data is converted into a numerical format that preserves the integrity of the original categories. This dual approach enhances the accuracy and efficacy of model training and analysis, ensuring that the categorical distinctions remain impactful without introducing artificial ordinal associations.

In the current phase of our research, the "Operator Label" values, categorized into "Good" and "Bad," were encoded numerically as 0 and 1, respectively, through Label Encoding. This numerical transformation is crucial for their effective incorporation into machine learning models, thereby enhancing prediction accuracy. Additionally, OneHot Encoding was applied to other categorical features within the dataset, including "ComponentID," "Result," "MachineID," and "AOILabel." This approach expanded the feature space from 20 to 120 dimensions, providing a more comprehensive representation for the analytical model. Furthermore, it was necessary to perform data-type conversions for several features initially typed as "object" within the dataset. To ensure full compatibility with the computational requirements of machine learning algorithms, these features—namely "ComponentID," "Result," "MachineID," and "AOILabel"—were converted to float data types. This conversion is vital for optimizing model performance and ensuring the robustness of subsequent analyses.

3.3.7 Feature Selection

Feature selection is a crucial preprocessing step in machine learning, particularly for managing datasets with high dimensionality and complexity. This process is essential for improving model effectiveness by reducing complexity, increasing accuracy, and decreasing the training time required. The primary goal of feature selection is to refine models for better clarity and to purify the dataset by removing redundant and irrelevant features (Jemai & Zarrad, 2023; Zingade, Deshmukh, & Kadam, 2023). Since the employed dataset in this study is suffering from class imbalance, Feature Selection can help balance class influence by eliminating redundant and irrelevant features that may bias the model toward the majority class. According to (Thiyam & Dey, 2024) by concentrating on features significant to both classes, feature selection enables the model to learn more effectively from minority class instances, thereby indirectly addressing class imbalance. In addition, as detailed in Section 3.2.3, we encountered numerous interrelated features within our dataset. Therefore, employing feature selection techniques, especially those that consider feature interaction and redundancy, is vital for improving the efficiency and effectiveness of machine learning models on datasets with interrelated features. The feature selection named Neighborhood Rough Set method, as demonstrated by (Wan et al., 2021), offers a powerful tool for achieving these benefits, leading to more accurate, interpretable, and resource-efficient models.

In the pursuit of refining data for enhanced analysis, several techniques are available for feature selection, each with distinct methodologies and advantages. Techniques such as PCA, Recursive Feature Elimination (RFE), LASSO (Least Absolute Shrinkage and Selection Operator), and Genetic Algorithms (GAs) are well-regarded in the field of data science for their efficacy in reducing dimensionality and isolating the most informative features, highlighting the potential of hybrid approaches that integrate multiple feature selection techniques (Luque-Rodriguez, Molina-Baena, Jimenez-Vilchez, & Arauzo-Azofra, 2022). This amalgamation harnesses the strengths of individual methods to provide a more robust solution for feature selection, thereby potentially improving model accuracy and interpretability. In related research, a two-stage feature selection process utilizing RFE with Logistic Regression and Gradient Boosting was explored to optimize the analytical

framework ([Mirzaei, 2023](#)). This approach has been shown to enhance model accuracy by effectively narrowing down crucial features and adeptly managing complex non-linear relationships. The hybrid method has proven superior, significantly refining feature selection and enhancing the robustness and adaptability of predictive models.

Building upon these insights, we employed a similar two-stage feature selection methodology using Recursive Feature Elimination coupled with Logistic Regression and Gradient Boosting, tailored with varying feature set sizes of 100, 80, and 60 to evaluate the impact of varying feature quantities on the predictive accuracy and robustness of the model. This strategy was chosen to leverage their combined strengths in accurately identifying and prioritizing features that significantly impact model outcomes. By methodically adjusting feature sets and employing a robust validation process, this approach aims to achieve superior predictive performance and reduce the likelihood of model overfitting, thus tailoring the model more precisely to the specific dynamics of the dataset at hand. The detailed findings, including the impact of different feature numbers, are provided in "Results and Discussion", Section [3.5.2](#).

3.3.8 Data Splitting

In machine learning, the practice of data splitting plays a pivotal role in building models that are both robust and capable of generalization. This process typically segregates the dataset into distinct training and testing subsets, essential for assessing a model's effectiveness on new, unseen data. Such separation is critical, preventing models from merely memorizing the data—a problem known as overfitting ([Reitermanova et al., 2010](#)).

In our work, the provided dataset was pre-partitioned into distinct training and testing sets. This preliminary separation ensures that any developed model can be validated on unseen data, thereby enhancing the reliability of its predictive capabilities. To facilitate validation, the training dataset was further segmented into a training set and a validation set, with careful attention to maintaining the original proportion of healthy and faulty classes. This division allows for the evaluation of models on data that was not used during training and enables early stopping and model tuning. Specifically, 20% of the total PCB panels were randomly selected to constitute the validation set, while the remaining 80% were designated for training. This balance was achieved by stratifying the

class labels, ensuring that the distribution of classes in the subsets mirrors that of the full dataset. After validation, the developed models were retrained on the entire training dataset to utilize all available data for learning. Finally, the separate test dataset, comprising 9,339 normal labels and 455 defective labels, was exclusively used to assess the performance of the models.

3.4 Model Implementation

Upon finalizing the data preparation phase, the subsequent step entailed the selection and training of suitable models utilizing our dataset. This section provides an overview of the machine learning and deep learning models employed, which are categorized into four distinct groups. The principal aim of employing these models is to explore the effects of synthetic data volume—generated via SMOTE and CTGAN techniques—on model performance. The section concludes with an analysis of the tools and libraries utilized throughout the model training process.

3.4.1 Instance-based Model

- **K-Nearest Neighbors (KNN):** This algorithm is a simple, non-parametric method used in classification and regression tasks. It works by identifying the K closest data points in the training set to a new data point and using these neighbors to predict the classification or value of the new point. KNN is particularly valued for its simplicity and effectiveness, especially when dealing with small datasets or cases where the relationship between features and labels is complex or non-linear. Although the algorithm is simple, selecting an appropriate value for K is crucial for accurately classifying unlabeled data ([Taunk, De, Verma, & Swetapadma, 2019](#)). Additionally, KNN has been successfully applied in diverse fields such as environmental geology and sentiment analysis, showing its versatility across different applications([Bullejos, Cabezas, Martín-Martín, & Alcalá, 2022](#)).

3.4.2 Tree-based Models

- **Random Forest:** The Random Forest model is a powerful ensemble learning method widely used in various fields for classification and regression tasks. It operates by constructing many decision

trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. This approach helps reduce overfitting and improve accuracy by leveraging multiple decision trees' diversity ([Cutler, Cutler, & Stevens, 2012](#)).

- **Extra Tree:** The Extra Trees (or Extremely Randomized Trees) algorithm is an ensemble machine learning method that aggregates the results of multiple de-correlated decision trees to improve the predictive accuracy and control over-fitting. The method differs from classic decision trees and random forests in the way it splits nodes. It uses random thresholds for each feature rather than searching for the best possible thresholds like a traditional decision tree does. This randomization typically increases the model variance reduction at the cost of a slight increase in bias, with the trade-off often leading to better model performance on complex datasets. Extra Trees can be particularly effective for large datasets and useful for classification and regression tasks ([Geurts, Ernst, & Wehenkel, 2006](#)).
- **Decision Tree:** The Decision Tree algorithm is a popular and powerful tool used for both classification and regression tasks in machine learning. It works by splitting the data into smaller subsets based on certain criteria, which are visualized as a tree structure. Each node in the tree represents a test on an attribute, each branch represents the outcome of that test, and each leaf node represents a class label (in classification) or a continuous outcome (in regression). The paths from root to leaf represent classification rules or regression paths. Decision trees are valued for their ease of interpretation and visualization, making them popular for exploratory data analysis. They can handle both numerical and categorical data and do not require data normalization. However, they can be prone to overfitting, especially with complex trees, but techniques such as pruning, setting the minimum number of samples required at a leaf node, or setting the maximum depth of the tree are commonly used to avoid this issue ([Breiman, 2017](#)).

3.4.3 Boosting-based Models

- **Gradient Boosting:** Gradient boosting is a versatile machine learning technique that incrementally constructs an ensemble of weak prediction models, typically decision trees, to enhance accuracy. Each new model in the ensemble sequentially corrects the errors of its predecessors by

focusing on the residuals, and continuously updating the loss function to minimize errors. This technique is flexible, supporting various loss functions and adaptable to both regression and classification tasks, making it widely applicable. Key parameters such as the number of trees, tree depth, and learning rate help control overfitting, allowing for a balanced approach between model complexity and generalization ([Friedman, 2001](#)). Gradient boosting is especially effective for structured data and remains popular in competitive machine-learning environments.

- **Light Gradient Boosting:** Light Gradient Boosting Machine is a fast and efficient gradient boosting framework developed by Microsoft that stands out for its use of a histogram-based algorithm which buckets continuous values into discrete bins to speed up training and reduce memory usage. It features Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) which improve model efficiency by focusing on more informative instances and bundling exclusive features, respectively. These innovations make LightGBM particularly suited for large datasets and scenarios demanding high computational performance ([Ke et al., 2017](#)).
- **XG Boost:** eXtreme Gradient Boosting is an advanced gradient boosting algorithm known for its efficiency and effectiveness in handling large and sparse datasets, developed by ([T. Chen & Guestrin, 2016](#)). XGBoost incorporates several innovative features such as L1 and L2 regularization to prevent overfitting, sparsity awareness that optimizes performance with missing data, and a weighted quantile sketch algorithm to manage weighted and imbalanced data effectively. It also supports parallel processing, enhancing its speed, and includes built-in cross-validation at each iteration, which aids in optimizing hyperparameters for improved model accuracy. It has been widely adopted in various data science applications due to its robustness and versatility.
- **Cat Boost:** Categorical Boosting is a powerful gradient boosting algorithm developed by Yandex, specifically optimized to handle categorical data seamlessly. Unlike other boosting algorithms that require extensive pre-processing to convert categorical values into numerical formats, CatBoost processes categorical features using its algorithmic approach, which reduces the risk of overfitting and enhances model performance. Key features of CatBoost include its use of ordered boosting to combat target leakage and improve generalization and symmetric trees that ensure balanced tree structures for more efficient predictions. This approach results in significant improvements in

handling categorical data, making CatBoost a preferred choice for models where categorical data is predominant (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018).

3.4.4 Hybrid/Deep Learning-based Model

- **TabNet:** TabNet is a novel deep learning architecture designed by (Arik & Pfister, 2021) that introduces a transformative approach to handling tabular data through a sequential attention mechanism. This architecture enables the model to selectively focus on the most salient features at each decision step, thereby optimizing both the interpretability and efficiency of learning. TabNet not only achieves superior performance across various datasets by using self-supervised learning to enhance model capabilities with unlabeled data, but it also offers both local and global interpretability. This dual-level interpretability—where local insights detail feature importance for individual predictions and global insights quantify overall feature contributions—makes TabNet particularly valuable in sectors like healthcare and finance, where understanding model reasoning is crucial. Building on this innovative design, TabNet has been adapted for specific applications that demonstrate its versatility and precision. For instance, in geosciences (Ta et al., 2023), it has been used to classify rock facies and extract feature embeddings from well-log data. Additionally, its application in remote sensing (Shah, Du, & Xu, 2022) for hyperspectral image classification highlights its effectiveness in managing spatial-temporal features.

TabNet’s architecture incorporates several innovative elements that enhance its performance in processing tabular data. It is designed with an emphasis on an attention mechanism, a decision-making process similar to that of decision trees, advanced feature transformation methods, and a tailored loss function to optimize training. These features collectively allow TabNet to identify and leverage complex nonlinear patterns within data, offering both scalability and interpretability.

1. Attention Mechanism: At the heart of TabNet is its attention mechanism which focuses selectively on the most crucial features at each decision step. This is achieved through the generation of attention weights w_i , which are assigned to each feature based on their importance. The attention weights are determined using the formula:

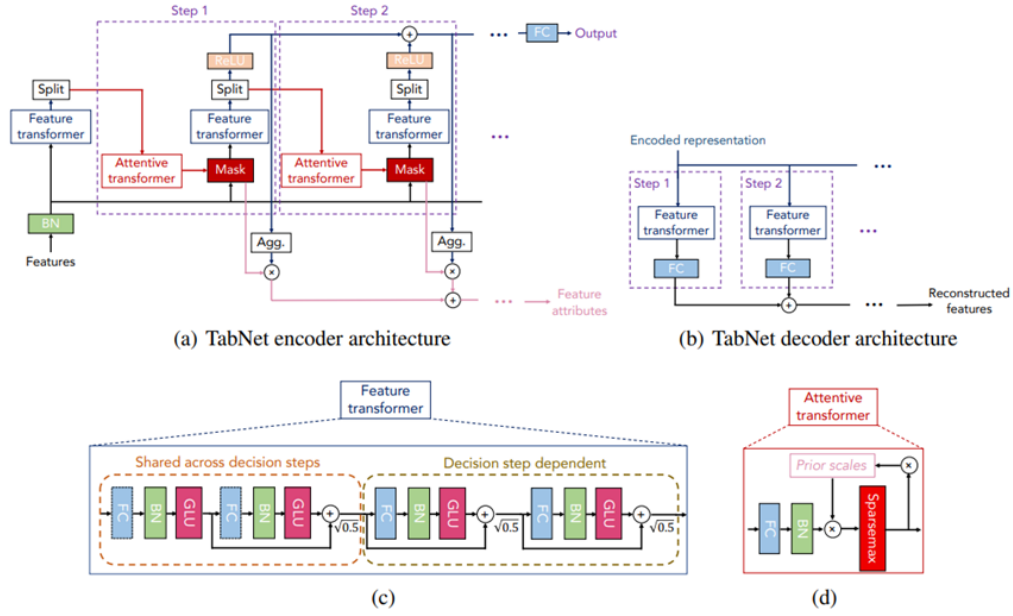


Figure 3.11: TabNet Model Architecture (Arik & Pfister, 2021)

$$w_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)} \quad (9)$$

Here, s_i represents a score reflecting the importance of the i -th feature, calculated through a trainable part of the network.

2. Split Decision: The split decision determines how the data is partitioned at each step of the decision process, mimicking the decision-making paths of traditional decision trees. It utilizes a specialized neural network layer to create splits in the data:

$$M = \text{Relu}(b + W \cdot \text{previous output}) \quad (10)$$

Where, M denotes the mask used to decide the data split, with W representing the weight matrix, b as the bias, and ReLU providing the necessary nonlinearity for learning intricate data patterns.

3. Feature Transformation: Before attention processing, TabNet often transforms features to enhance their compatibility with the model structure. This transformation typically involves a

linear adjustment followed by a nonlinear activation function. This could be represented by:

$$F = \text{Relu}(W \cdot x + b) \quad (11)$$

In this equation, F stands for the transformed feature, x is the input feature, with W and b as the transformation's weights and bias, respectively.

4. Loss Function: The training of TabNet involves minimizing a specific loss function that reflects how closely the model's predictions match the actual data labels. In binary classification tasks, this function could be the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (12)$$

where N is the total number of samples, y_i is the true label, and \hat{y}_i is the predicted label by the model.

These components make TabNet a powerful tool in the arsenal of modern machine learning, capable of handling a variety of tasks with high efficiency and interpretability.

3.4.5 Tools and Libraries

In this study, machine learning model development was facilitated by using *PyCaret* ([PyCaret — pycaret 2.3.5 documentation, n.d.](#)), an accessible, low-code library in Python designed for machine learning automation. This platform streamlines numerous tasks within the machine learning pipeline such as data preprocessing, model training, hyperparameter optimization, and model evaluation. The choice of PyCaret was motivated by its user-friendly interface and its efficiency in experimenting with various models and techniques.

Additionally, for the deployment of the TabNet model, the *PyTorch* ([PyTorch, n.d.](#)) framework was employed. PyTorch, known for its powerful computing capabilities and flexibility, supports detailed model architecture customization and comprehensive control over the training process. The TabNet model, which is tailored for structured data, benefits from the extensive features of PyTorch,

allowing for precise adjustment of the model’s parameters to enhance performance.

3.5 Experimental Results

In this section, we begin by defining the essential metrics required for the accurate evaluation and refinement of our ML/DL models. The selection of appropriate metrics is paramount, as choosing unsuitable ones can result in misleading interpretations of model performance. Acknowledging the importance of this selection, we will explore both Class-level and Model-level metrics in detail. Furthermore, we thoroughly investigate the impact of varying volumes of synthetic data—generated using SMOTE and CTGAN techniques—on the performance of our proposed models. This analysis is conducted separately for each technique, aiming to understand how different quantities of synthetic data influence model efficacy. By examining these effects in detail, we aim to provide insights into optimizing machine learning workflows through data augmentation and identify the most effective strategies for enhancing model performance within the context of this project.

3.5.1 Metrics and Indicators

Choosing the right metrics to evaluate our model depends critically on the characteristics of our dataset and the goals we aim to achieve. As previously discussed in Section 3.1, our primary objective in this chapter is to classify defects effectively along the PCB operator label to minimize subsequent costs. Given this focus, class-level metrics are especially pertinent due to the high cost associated with false negatives. These metrics help us gauge the reliability of our model in identifying defective components, which is crucial for preventing costly errors.

Moreover, model-level metrics are crucial in situations involving imbalanced data sets or when different errors carry different costs. In our case, incorrectly classifying a functional PCB component as defective can lead to unnecessary expenses, either through further inspections or by discarding a functional board. Therefore, model-level metrics provide a holistic evaluation of our model’s performance by considering various aspects of its accuracy and the implications of errors. Further details on these metrics and their application are discussed below.

(1) Class-level Metrics

Class-level metrics offer a detailed assessment of model performance by measuring how accurately it predicts each class (defective and non-defective) (B. J. Erickson & Kitamura, 2021). These metrics, along with their formulas, are outlined below:

- **Precision:** This is calculated as the ratio of true positive predictions to the total number of positive predictions. It reflects the proportion of predicted positive instances that are correctly identified.

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

- **Recall:** This is determined by the ratio of true positive predictions to the total number of actual positive instances. It assesses the model's effectiveness in identifying all positive cases.

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

- **F1 Score:** The F1 score is the harmonic mean of precision and recall, offering a balanced measure that is particularly useful for datasets with class imbalance. The closer it is to 1, the better the model's performance in accurately classifying the minority class.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (15)$$

(2) Model-level Metrics

Model-level metrics provide an overall evaluation of the model's performance. These metrics include:

- **Accuracy:** The ratio of correctly predicted instances to the total number of instances. While accuracy provides a simple measure of performance, it may not be sufficient for imbalanced datasets.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

- **AUC:** The area under the ROC curve (AUC) quantifies the model's discriminatory power, with values nearer to 1 indicating better performance. The "receiver operating characteristic

(ROC) curve” displays the results of a model’s classification predictions by plotting the true positive rate (recall) on the vertical axis against the false positive rate (FPR, which is one minus the specificity) on the horizontal axis (Eom & Byeon, 2023). AUC is computed as the average of the true positive rate and the true negative.

$$Specificity = \frac{TN}{TN + FP} = TNR \quad (17)$$

$$AUC = \frac{TPR + TNR}{2} \quad (18)$$

3.5.2 Results and Discussion

In this section, we assess the impact of varying volumes of synthetic data, generated through SMOTE and CTGAN techniques, on the efficacy of the models delineated in Section 3.4. Given the pronounced imbalance within our dataset, where classifiers typically yield high accuracy yet low recall, the F1-score—particularly with an emphasis on the minority class representing defective PCBs—serves as a critical metric for evaluating classifier performance. Additionally, each algorithm is trained using 5-fold cross-validation to ensure reliability and robustness.

Our analysis commenced with KNN model, leveraging the SMOTE and CTGAN augmentation methods. As observed in Tabel 3.6, the F1-score on the test set was initially 42% without the use of balancing techniques. Further exploration into the impact of synthetic data volume on the F1-score demonstrated an improvement to 50% after augmenting the dataset by 40% with SMOTE-generated synthetic data. When deploying CTGAN-generated data, as illustrated in Tabel 3.7, a threshold augmentation of 60% improved the F1-score to 54%. However, any additional increase in synthetic data proved counterproductive, leading to a decrement in model performance.

Table 3.6: KNN Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.984±0.001	0.902±0.013	0.903±0.034	0.657±0.031	0.76±0.023
		Test	0.959	0.723	0.62	0.323	0.424
0.2	4310	Valid	0.972±0.003	0.98±0.004	0.939±0.011	0.889±0.021	0.913±0.011
		Test	0.956	0.769	0.529	0.472	0.497
0.4	8620	Valid	0.972±0.002	0.989±0.002	0.954±0.005	0.938±0.004	0.946±0.004
		Test	0.952	0.774	0.489	0.500	0.499
0.6	12930	Valid	0.970±0.003	0.992±0.001	0.96±0.005	0.960±0.004	0.960±0.005
		Test	0.946	0.773	0.435	0.503	0.466
0.8	17240	Valid	0.971±0.003	0.993±0.0009	0.964±0.005	0.97±0.003	0.967±0.002
		Test	0.942	0.77	0.406	0.518	0.455
Minority	21551	Valid	0.974±0.003	0.993±0.0008	0.967±0.003	0.980±0.003	0.974±0.001
		Test	0.94	0.769	0.393	0.516	0.446

Table 3.7: KNN Performance Metrics at Different Levels of CT-GAN Synthetic Data

Imbalance Technique	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.984±0.001	0.902±0.013	0.903±0.034	0.657±0.031	0.760±0.023
		Test	0.9594	0.723	0.62	0.323	0.424
	4310	Valid	0.976±0.003	0.965±0.006	0.979±0.009	0.877±0.017	0.925±0.010
		Test	0.962	0.765	0.64	0.435	0.518
	8620	Valid	0.973±0.002	0.977±0.003	0.990±0.003	0.915±0.011	0.95±0.005
		Test	0.963	0.777	0.657	0.439	0.527
CT-GAN	12930	Valid	0.971±0.003	0.983±0.002	0.992±0.003	0.930±0.006	0.960±0.004
		Test	0.963	0.779	0.66	0.45	0.54
	17240	Valid	0.971±0.001	0.986±0.002	0.994±0.001	0.941±0.004	0.967±0.002
		Test	0.964	0.777	0.671	0.448	0.537
	21551	Valid	0.97±0.002	0.987±0.001	0.995±0.002	0.845±0.006	0.969±0.003
		Test	0.964	0.776	0.674	0.446	0.537

Table 3.8: LGBM Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.987±0.001	0.947±0.012	0.933±0.024	0.715±0.027	0.812±0.02
		Test	0.960	0.793	0.691	0.285	0.404
0.2	4310	Valid	0.982±0.002	0.990±0.004	0.979±0.007	0.915±0.017	0.946±0.007
		Test	0.963	0.775	0.713	0.367	0.484
0.4	8620	Valid	0.981±0.002	0.994±0.001	0.985±0.004	0.947±0.007	0.966±0.005
		Test	0.964	0.757	0.709	0.391	0.504
0.6	12930	Valid	0.979±0.003	0.996±0.001	0.989±0.003	0.956±0.007	0.972±0.004
		Test	0.961	0.755	0.644	0.378	0.476
0.8	17240	Valid	0.979±0.002	0.996±0.001	0.989±0.003	0.963±0.005	0.976±0.003
		Test	0.962	0.749	0.648	0.408	0.501
Minority	21551	Valid	0.979±0.003	0.997±0.0006	0.99±0.003	0.968±0.004	0.979±0.003
		Test	0.959	0.748	0.597	0.384	0.467

A parallel evaluation of the LightGBM model, with a random state of 123, echoed these findings. This model demonstrated optimal results with a 40% enhancement using SMOTE data and a 60% increase with CTGAN-generated data, as detailed in the respective Tables 3.8, and 3.9. When comparing with the KNN model, LightGBM displayed a 4% decrease in performance with CTGAN sourced data.

Table 3.9: LGBM Performance Metrics at Different Levels of CT-GAN Synthetic Data

Imbalance Technique	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.987 ± 0.001	0.947 ± 0.012	0.939 ± 0.027	0.715 ± 0.024	0.812 ± 0.020
		Test	0.960	0.793	0.692	0.285	0.404
	4310	Valid	0.987 ± 0.002	0.987 ± 0.005	0.988 ± 0.006	0.936 ± 0.013	0.962 ± 0.006
		Test	0.960	0.811	0.671	0.305	0.419
	8620	Valid	0.989 ± 0.001	0.994 ± 0.001	0.993 ± 0.002	0.979 ± 0.003	0.987 ± 0.002
		Test	0.960	0.809	0.656	0.327	0.437
CT-GAN	12930	Valid	0.990 ± 0.001	0.996 ± 0.012	0.995 ± 0.027	0.715 ± 0.024	0.812 ± 0.020
		Test	0.962	0.834	0.688	0.340	0.455
	17240	Valid	0.992 ± 0.003	0.997 ± 0.006	0.997 ± 0.009	0.985 ± 0.017	0.991 ± 0.010
		Test	0.960	0.833	0.646	0.329	0.436
	21551	Valid	0.992 ± 0.001	0.997 ± 0.0007	0.997 ± 0.001	0.987 ± 0.002	0.992 ± 0.001
		Test	0.961	0.818	0.682	0.320	0.436

Furthermore, trials involving the XGBoost model (See Tabel 3.10, and 3.11), reaffirmed the limitations of excessive synthetic data augmentation. Specifically, a threshold of 40% in synthetic data augmentation was identified beyond which no further improvements in model performance were discernible.

Table 3.10: XGBoost Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1.Score
None	858	Valid	0.987±0.001	0.944±0.012	0.945±0.020	0.724±0.029	0.819±0.020
		Test	0.960	0.782	0.697	0.252	0.371
0.2	4310	Valid	0.983±0.003	0.991±0.003	0.979±0.006	0.922±0.020	0.949±0.010
		Test	0.963	0.78	0.71	0.3451	0.464
0.4	8620	Valid	0.982±0.003	0.995±0.001	0.986±0.004	0.951±0.007	0.968±0.005
		Test	0.962	0.770	0.674	0.364	0.473
0.6	12930	Valid	0.982±0.003	0.997±0.001	0.988±0.003	0.963±0.005	0.975±0.004
		Test	0.962	0.761	0.676	0.363	0.473
0.8	17240	Valid	0.982±0.002	0.997±0.005	0.988±0.003	0.970±0.005	0.979±0.002
		Test	0.961	0.764	0.648	0.36	0.463
Minority	21551	Valid	0.982±0.002	0.998±0.005	0.990±0.002	0.973±0.005	0.982±0.002
		Test	0.960	0.761	0.609	0.386	0.473

Table 3.11: XGBoost Performance Metrics at Different Levels of CT-GAN Synthetic Data

Imbalance Technique	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.987 ± 0.001	0.944 ± 0.012	0.945 ± 0.029	0.726 ± 0.020	0.819 ± 0.020
		Test	0.960	0.782	0.697	0.252	0.371
	4310	Valid	0.987 ± 0.002	0.987 ± 0.004	0.987 ± 0.006	0.936 ± 0.014	0.961 ± 0.008
		Test	0.959	0.810	0.656	0.285	0.398
	8620	Valid	0.989 ± 0.001	0.994 ± 0.001	0.993 ± 0.003	0.970 ± 0.005	0.993 ± 0.002
		Test	0.961	0.807	0.650	0.338	0.446
	12930	Valid	0.990 ± 0.001	0.996 ± 0.001	0.996 ± 0.002	0.979 ± 0.003	0.987 ± 0.002
		Test	0.961	0.805	0.679	0.307	0.423
	17240	Valid	0.991 ± 0.001	0.997 ± 0.0008	0.996 ± 0.001	0.985 ± 0.002	0.990 ± 0.001
		Test	0.960	0.823	0.648	0.307	0.417
	21551	Valid	0.993 ± 0.001	0.997 ± 0.0006	0.997 ± 0.001	0.988 ± 0.002	0.992 ± 0.001
		Test	0.960	0.790	0.653	0.314	0.424

To corroborate our results further, we extended our validation efforts to encompass additional machine learning models such as Extra Trees, Random Forest, Decision Tree, Gradient Boosting, and CatBoost. The consistency in results across these models validated the efficacy of our synthetic data volume threshold. The comprehensive details and tables of these findings are meticulously cataloged in the Appendix [A](#).

Transitioning to the deep learning spectrum, the performance metrics of the TabNet model are delineated in Tables [3.12](#) and [3.13](#). Throughout the model deployment phase, Grid Search methodology was employed to meticulously fine-tune TabNet’s hyperparameters, ensuring optimal configuration tailored to the dataset’s unique characteristics. The optimal hyperparameters were identified as follows: number of decision steps (n_d) at 63, number of attention units (n_a) at 40, a step size

Table 3.12: TabNet Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Train	0.949±0.006	0.966±0.010	0.427±0.034	0.882±0.019	0.574±0.031
		Valid	0.947±0.006	0.939±0.008	0.409±0.029	0.829±0.028	0.547±0.022
		Test	0.926	0.783	0.316	0.507	0.389
0.2	4310	Train	0.973±0.004	0.994±0.001	0.898±0.020	0.946±0.008	0.921±0.013
		Valid	0.964±0.006	0.986±0.003	0.873±0.023	0.923±0.018	0.897±0.017
		Test	0.950	0.703	0.454	0.452	0.453
0.4	8620	Train	0.972±0.003	0.995±0.0007	0.951±0.010	0.953±0.001	0.952±0.005
		Valid	0.968±0.003	0.992±0.001	0.942±0.008	0.946±0.006	0.944±0.005
		Test	0.951	0.640	0.473	0.393	0.429
0.6	12930	Train	0.967±0.006	0.994±0.002	0.962±0.011	0.949±0.010	0.955±0.009
		Valid	0.963±0.006	0.992±0.002	0.956±0.012	0.945±0.012	0.950±0.009
		Test	0.942	0.645	0.381	0.375	0.378
0.8	17240	Train	0.969±0.001	0.995±0.0006	0.976±0.004	0.955±0.008	0.965±0.002
		Valid	0.966±0.003	0.994±0.001	0.970±0.006	0.952±0.009	0.961±0.003
		Test	0.944	0.667	0.401	0.415	0.408
Minority	21551	Train	0.970±0.004	0.996±0.001	0.986±0.003	0.953±0.009	0.969±0.004
		Valid	0.966±0.004	0.994±0.001	0.982±0.004	0.951±0.011	0.966±0.005
		Test	0.951	0.678	0.470	0.417	0.442

of 10, a decay rate (gamma) of 0.84, a learning rate (lr) of 0.001, batch size of 256, and virtual batch size of 128, with 'entmax' selected as the mask type. An early stopping mechanism with a patience setting of 10 epochs was also implemented to prevent overfitting, thus enhancing the model's stability and generalization capabilities.

Regarding performance, without any synthetic data augmentation, TabNet achieved an F1-score of 54% on the validation set and approximately 39% on the test set, highlighting its stability in contrast to traditional machine learning models. With a modest 20% increase in SMOTE data, the optimal F1-score was observed at 45%. The model's ability to discriminate between classes also improved, with the AUC metric rising from 78% to 83%, as depicted in Figure 3.12. This multi-model validation approach not only underscores the generalizability of our conclusions across different frameworks but also emphasizes the nuanced balance required in synthetic data utilization to optimize model performance within PCB production line datasets.

Table 3.13: TabNet Performance Metrics at Different Levels of CT-GAN Synthetic Data

Imbalance Technique	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Train	0.949±0.006	0.966±0.010	0.427±0.034	0.882±0.019	0.574±0.031
		Valid	0.947±0.006	0.939±0.008	0.409±0.029	0.829±0.028	0.547±0.022
		Test	0.926	0.783	0.316	0.507	0.389
	4310	Train	0.986 ± 0.009	0.990 ± 0.001	0.989 ± 0.004	0.930 ± 0.005	0.959 ± 0.002
		Valid	0.984 ± 0.001	0.987 ± 0.003	0.984 ± 0.002	0.919 ± 0.009	0.950 ± 0.004
		Test	0.962	0.759	0.679	0.353	0.465
	8620	Train	0.988 ± 0.001	0.995 ± 0.001	0.994 ± 0.002	0.964 ± 0.004	0.979 ± 0.002
		Valid	0.985 ± 0.001	0.993 ± 0.001	0.990 ± 0.004	0.958 ± 0.006	0.974 ± 0.002
		Test	0.964	0.765	0.753	0.356	0.483
	CT-GAN 12930	Train	0.990 ± 0.008	0.996 ± 0.003	0.996 ± 0.001	0.978 ± 0.001	0.987 ± 0.001
		Valid	0.988 ± 0.002	0.995 ± 0.009	0.995 ± 0.001	0.973 ± 0.005	0.984 ± 0.002
		Test	0.96	0.827	0.634	0.351	0.452
	17240	Train	0.992 ± 0.0009	0.997 ± 0.0002	0.997 ± 0.001	0.985 ± 0.001	0.991 ± 0.001
		Valid	0.990 ± 0.0006	0.997 ± 0.0004	0.995 ± 0.0008	0.982 ± 0.001	0.989 ± 0.0007
		Test	0.960	0.776	0.645	0.351	0.455
	21551	Train	0.992 ± 0.001	0.998 ± 0.0004	0.997 ± 0.001	0.986 ± 0.001	0.992 ± 0.001
		Valid	0.990 ± 0.001	0.997 ± 0.0002	0.996 ± 0.001	0.983 ± 0.001	0.989 ± 0.001
		Test	0.960	0.758	0.662	0.340	0.449

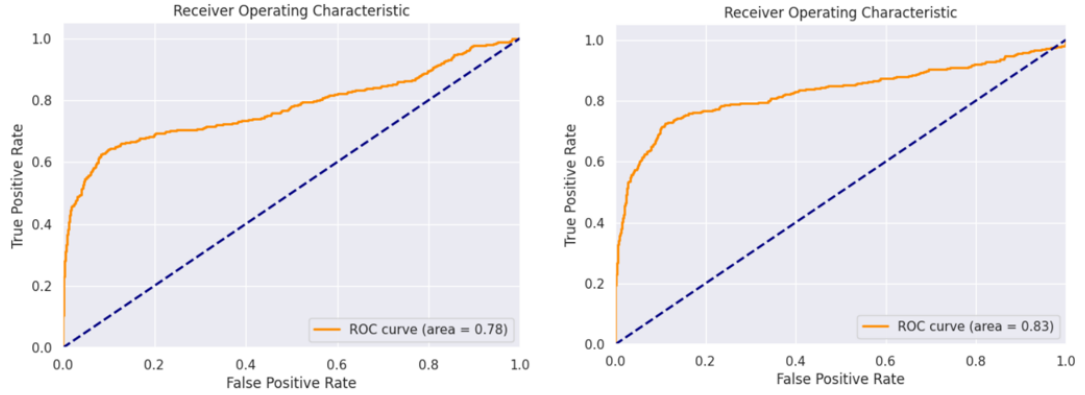


Figure 3.12: ROC curves for TabNet Model

According to sensitivity analysis from the test dataset, among the ML algorithms, the KNN classifier demonstrated superior performance, achieving an F1-score of 54%. In comparison, the deep learning-based TabNet model achieved an F1-score of 48%, both with 40% of the data augmented using CT-GAN. The relatively low performance of the models on the test dataset can primarily be

attributed to an imbalance in the class distribution within the test data.

Oversampling was a critical component of our data preprocessing strategy. However, a significant risk associated with this method is the potential for overfitting. Consequently, there was a noticeable disparity between the performance metrics during the training and testing phases, attributable to the different balance ratios in these datasets.

3.5.3 Comparing Our Approaches to Similar Solutions

As we explore the intricacies of ML and DL techniques in PCB manufacturing, it is essential to contextualize our methods within the broader landscape of existing research. This section provides a comparative analysis, aimed at demonstrating how our work aligns with, and diverges from, parallel studies in the field. By examining the key similarities and differences, we aim to underscore the distinct contributions and advantages of our approach.

Table 3.14 provides a comparative analysis of F1-scores for Operator Label prediction across our methods and four related studies. Our KNN model, which achieved an F1-score of 0.96 during training and 0.54 during testing, demonstrates strong learning capabilities with comparable test performance to Case 3 (Tang et al., 2022). Although our TabNet model achieved the highest F1-score during training (0.97), its performance dropped to 0.48 on the test set, due to ongoing label imbalance, similar to the performance of Case 2 (Gore et al., 2022).

Table 3.14: F1-score Comparison for OperatorLabel Prediction

Comparison Cases	Train	Test
Our Method (TabNet Result)	0.97	0.48
Our Method (KNN Result)	0.96	0.54
Case 1 (Gaffet, Roa, Ribot, Chanthery, & Merle, 2022)	0.66	0.67
Case 2 (Gore et al., 2022)	0.71	0.48
Case 3 (Tang et al., 2022)	0.68	0.54
Case 4 (Schmidt, Dingeldein, Hünemohr, Simon, & Weigert, 2022)	0.81	0.38

However, despite this drop in performance, our approach strategically utilizes a **40%** CTGAN

oversampling threshold, which remains competitive and particularly advantageous. This methodology not only sustains robust training performance but also effectively manages the challenges posed by the highly imbalanced dataset. This suggests that with further refinements, such as additional regularization or adjusted data preprocessing techniques, our models could close the gap between training and testing performance more effectively. Our approach, therefore, offers a promising foundation that aligns with existing methodologies' ability to learn from and generalize across diverse PCB data.

3.6 Summary of The Chapter

This chapter focuses on automating the quality control process in PCB manufacturing, particularly during the AOI stage, by utilizing machine learning and deep learning models for defect detection. It also compares the effectiveness of these models with human operators, highlighting how ML and DL models offer significant advantages in speed, consistency, and reducing human error. It details the steps involved in data preparation, including data cleaning, aggregation, and handling imbalanced data, and introduces various ML and DL models such as instance-based models, tree-based models, boosting-based models, and TabNet model. The implementation of these models is discussed with a focus on optimizing performance through synthetic data augmentation, particularly using techniques like SMOTE and CTGAN.

The experimental results are presented and analyzed in depth, with the performance of the models evaluated using various metrics. The chapter discusses the impact of synthetic data volume on model performance, highlighting the trade-offs between different levels of data augmentation. A comparison is made between the proposed approaches and existing solutions in the literature, emphasizing the advantages of the proposed methods, particularly in handling data imbalances and improving the accuracy of PCB defect detection. The chapter concludes by summarizing the key findings, reiterating the importance of synthetic data volume in enhancing the performance of ML and DL models in PCB defect detection.

Chapter 4

Prediction of Human Repair Labels in PCBs: Leveraging Feature Engineering and Ensemble Learning Techniques

In the preceding chapter, we explored the effects of imbalance handling techniques and synthetic data volumes on the predictive accuracy of machine learning and deep learning models in assessing the health status of PCBs post-component replacement. This assessment is vital for determining the boards' suitability for subsequent manufacturing stages, where PCBs in good condition are advanced in the production line, while defective ones are earmarked for repair or disposal. Building on these insights, this chapter seeks to further optimize PCB manufacturing by automating the assignment of repair labels using data analytics and ML models.

The traditional process involves manually inspecting PCBs labeled "Bad" during the "OperatorLabel" stage to determine their repair potential. Automating this evaluation process could significantly enhance the efficiency and accuracy of decisions, thus reducing dependency on manual intervention. By implementing an automated repair label generation system that leverages advanced ML algorithms, we aim to accelerate the production process, ensure consistent and precise board evaluations, and minimize labor and material costs. The data-driven nature of machine learning also facilitates continuous improvements and compliance with stringent manufacturing standards,

offering an efficient, precise, and cost-effective solution for PCB quality control.

The remainder of this chapter is structured as follows: Section 4.1 introduces the problem under investigation. Section 4.2 details the data preprocessing and preparation steps undertaken. This section also explains the feature engineering processes involved. Section 4.3 discusses the ML models used for "RepairLabel" classification and the specific libraries used for implementation. Section 4.4 evaluates the performance metrics of ML models, and reviews the implementation process, outcomes, and feature importance. It concludes with a comparative analysis of the models, discussing their strengths and weaknesses. Finally, Section 4.5 provides a summary of the chapter, encapsulating the major insights and contributions of the research.

4.1 Problem Identification

In the domain of PCB manufacturing, the development of robust repair strategies and precise predictive labeling of repair needs are vital for economic and environmental sustainability. Repairing items instead of replacing them not only reduces operational costs by extending the lifespan of the boards but also maintains production efficiency critical for industries facing rapid market evolution such as automotive, aerospace, and consumer electronics. This practice also significantly reduces electronic waste, thereby aiding environmental conservation. Accurate repair labeling is crucial for optimizing resource use and promoting sustainable manufacturing practices. However, the process of determining the specific repair requirements of PCBs is complex and traditionally relies on expert analysis, which can be error-prone and inconsistent.

To address these challenges, this chapter introduces a predictive model designed to accurately assign "RepairLabel" to the products after a detailed post-inspection analysis. PCBs identified with defects are routed to a specialized section for a reparability assessment conducted by experts. This manual process is fraught with potential errors due to the intricate nature of components and the high dependency on the operator's expertise. The proposed model utilizes ML techniques, employing predictive variables from SPI and AOI datasets to forecast the repair requirements of PCBs accurately. The model categorizes products into irreparable, or incorrectly marked as scrap, enhancing the decision-making process in PCB repair and ultimately improving the efficiency and

sustainability of manufacturing operations.

4.2 Data Preprocessing

This section delineates the comprehensive methodology applied to data preprocessing, feature engineering, and the evaluative measures implemented during model training. The objective is to clearly outline the processes involved in converting raw data into a structured format amenable to predictive modeling, thereby facilitating the extraction of actionable insights from the datasets. Consistent with the approaches described in Chapter 3 (Section 3.3), the data utilized in this study—including the SPI and AOI datasets—were sourced, merged, and prepared, maintaining the same dataset size as previously documented. However, this chapter introduces a different approach to data preprocessing, placing a heightened emphasis on feature engineering designed to bolster our predictive analysis concerning repair label assignments. The subsequent sections will provide an in-depth exploration of these refined methods.

4.2.1 Data preparation and Cleaning

In this preparation phase, we adopted an approach similar to the one outlined in Section 3.3.1, providing a unified SPI-AOI dataset comprising 25 attributes. In contrast to the previous chapter, features such as PanelID, Date, and Time are preserved to be employed for the feature engineering step. Moreover, we have integrated an additional feature, "OperatorLabel," into our dataset. Initially utilized as a target for prediction, this feature has been reclassified as an input variable to leverage its significant impact on model decision-making processes.

As detailed in Table 3.2, the target column "RepairLabel" consists of four classes, including NA (Not Available), NotYetClassified, NotPossibleToRepair, and FalseScrap. "NA," labels indicate components previously identified as "Good" and thereby exempt from further repair scrutiny, reducing its total size to 903 rows. This refinement process not only preserves data integrity but also enhances the quality of the dataset. Considering that our primary objective is to determine whether our products are mislabeled as scrap or are truly irreparable, the target column "RepairLabel" included entries labeled as "NotYet Classified," which pose ambiguity regarding their condition. The

presence of such indeterminate labels in the dataset could potentially introduce noise and compromise the accuracy of our predictive analysis. To mitigate this risk, entries with "NotYet Classified" labels were excluded from the dataset. This exclusion is crucial as it prevents the model from learning patterns associated with these non-informative labels, thereby enhancing the overall quality and reliability of the predictive model. The distribution of the remaining target labels within the training and testing datasets is visually depicted in Figure 4.1.

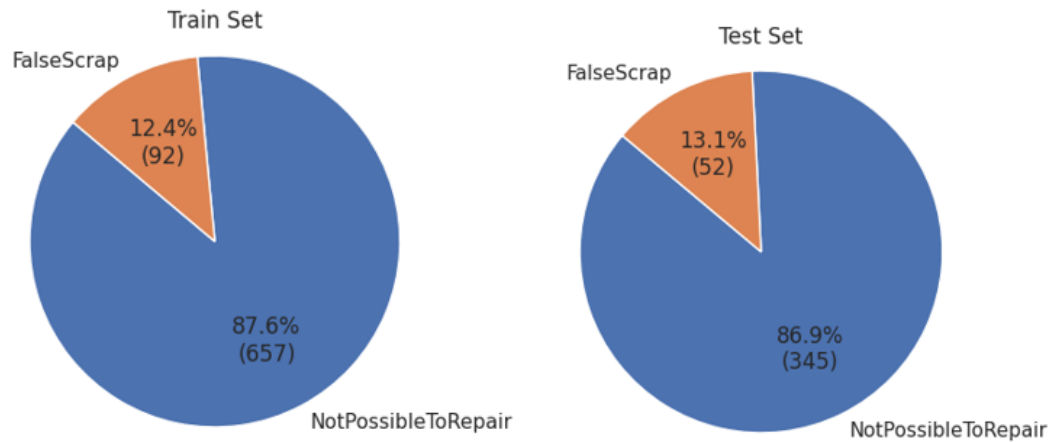


Figure 4.1: Repair Labels Distribution

4.2.2 Data Type Standardization

A key step in preparing our data was to standardize the data types across all features in the training and testing datasets. In this regard, 'FigureID' and 'PinNumber,' which were originally in different formats, both converted to integers.

Following the guidelines in Section 3.3.3, we also took careful steps to maintain the quality of our training and testing data by addressing any inconsistencies in the 'ComponentID' feature between the two sets. To ensure that our models would be tested on completely new, unseen data, we removed any ComponentIDs that not shared between the datasets. Specifically, we removed 12 entries—9 from the training set and 3 from the testing set. This step is crucial to prevent any overlap, which is essential for avoiding data leakage that could bias our results. These measures

help ensure that our evaluation of the model’s performance is both fair and accurate.

4.2.3 Feature Engineering

Feature engineering is a critical process in the analysis of SPI and AOI datasets in PCB circuit boards. This process involves transforming raw data into meaningful features that can enhance model robustness and accuracy. Effective feature extraction is essential, as it can significantly improve the predictive performance of machine learning models by emphasizing the most relevant characteristics of the data. Several studies have highlighted the importance of feature engineering in the context of PCB inspection and fault detection. For instance, (Tang et al., 2022) demonstrated that carefully crafted statistical features derived from SPI data could substantially improve model performance in detecting PCB defects. Similarly, (Gore et al., 2022) explored the pivoting technique for feature extraction, showing that the integration of domain-specific knowledge in feature engineering leads to more accurate and reliable predictions. The robustness of models is highly dependent on the quality of features extracted from the data. Enhanced features facilitate early anomaly detection, reducing waste and improving yield in PCB manufacturing, thus highlighting the critical role of feature engineering in optimizing the quality control process (I.-C. Chen, Hwang, & Huang, 2023).

In our approach, we Initially identified key features and then applied various transformation techniques to enhance the dataset’s predictive power. We also extracted new features, considering feature interactions and aggregations, to capture the underlying patterns more effectively. The detailed methodology for this process is explained in the subsequent sections of this chapter.

- (1) **Feature Transformation:** Given the presence of various categorical features in our dataset, as detailed in Section 3.3.6 (Data Encoding), transforming these features is crucial since most machine learning algorithms require numerical input to function effectively. The presence of high cardinality within categorical columns like "ComponentID" demands an in-depth understanding of their structure and composition. In our dataset, this column lists identifiers for different components on PCB circuit boards, such as C12, D1, R11, and TR2. Here, the

initial letter indicates the type of component, and the subsequent number denotes the number of pins. To manage this complexity and boost analytical efficiency, we categorized these identifiers into distinct groups based on their globally recognized symbols, using a regular expression function. This function assigns each ComponentID to a category such as Connector, Transistor, Capacitor, Resistor, IC (Integrated Circuit), Diode, Inductor, and Other. This preprocessing step introduced a single new column, '*Component_Category*', containing eight unique categories, providing a clear and concise classification of each component.

Further processing involved one-hot encoding technique to this new column, as well as the '*AOILabel*', to numerically encode the various labels, ensuring compatibility with machine learning algorithms. Additionally, the '*OperatorLabel*' and '*RepairLabel*' were transformed into a binary format using label encoding. In this process, entries labeled as 'Good' in OperatorLabel and 'FalseScrap' in the RepairLabel were assigned a value of 1, while Bad and NotPossibleToRepair were assigned a value of 0.

Moreover, the 'Result' column was processed using a new function, which assigns numerical labels from 1 to 5 based on the type of result and stores the results in a new feature named '*ResultCategory*'. In addition, a new '*Result_Binary*' column was also created where 'Good' labels, which constitute the majority, were specifically encoded as 1 and all other few defective labels as 0. This dual approach of label encoding and binary conversion optimizes the target features for machine learning applications, significantly enhancing the accuracy and efficiency of subsequent analyses.

(2) **Feature extraction:** This step involves generating new variables and performing geometric feature extraction using basic mathematical operations, which can help quantify complex relationships between features. Each new feature is developed by leveraging significant interactions among existing variables. The steps taken are outlined as follows:

- **Height_SizeX and Height_SizeY:** These two features are created by multiplying the height with its dimensions in the X and Y directions, respectively. This captures the volumetric properties of the solder paste, which are critical for identifying defects related to insufficient or excessive solder application.

- **AspectRatio:** This feature is calculated by dividing SizeX by SizeY, providing valuable insight into the geometric properties and shape of a PCB component. This metric is crucial for predicting and preventing defects related to improper shape or alignment.
- **Distance_from_origin:** This feature calculates the Euclidean distance from the origin to the position of the solder paste, indicating its spatial placement on the PCB. It is useful for identifying placement errors and misalignments. The formula applied is as follows:

$$Distance_from_origin = \sqrt{(\text{PosX}(\text{mm}))^2 + (\text{PosY}(\text{mm}))^2} \quad (19)$$

By implementing these simple mathematical operations, we are able to extract meaningful interactions between features without relying on unnecessary logarithmic transformations, helping the model gain a better understanding of the data and leading to improved predictive performance.

- (3) **Feature Aggregation:** In this phase, which predominantly involves statistical features, we focused on deriving new features by leveraging various statistical properties. We selected three key identifiers—‘*PanelID*’, ‘*FigureID*’, and ‘*ComponentID*’—as relevant grouping variables. Using these identifiers as a group, we integrated three statistical measures, mean, maximum, and minimum, across several key pin-level features such as ‘*Area (um²)*’, ‘*Shape (um)*’, ‘*Volume (um³)*’, ‘*Height (um)*’, ‘*OffsetX (%)*’, and ‘*OffsetY (%)*’. This step enabled us to derive 18 new features that represent the overall characteristics of the components within each group. These features provide a comprehensive statistical summary of the components, enhancing our model’s ability to predict outcomes more accurately. The importance of this approach and its contributions to our model’s effectiveness will be further discussed and demonstrated in Section 4.4.3, where we will highlight how these aggregated features play a pivotal role in our predictive analysis.

(4) **Domain-Specific Features:** Incorporating domain-specific knowledge into feature engineering is crucial for developing sophisticated features that are specifically tailored to PCB manufacturing. Recognizing that certain defects are more common in PCB layouts can significantly inform this process. As demonstrated earlier in Tabel 3.2, 'AOILabel' variable includes several labels categorizing types of defects. Focusing on issues such as coplanarity, misalignment, and solder characteristics and fostering domain-specific knowledge, we aimed to generate new features to capture nuanced aspects of component quality and defects.

- **Coplanarity:** In PCB manufacturing, coplanarity refers to the degree to which the surfaces of components, particularly their pins, are level with one another. It's a critical measure in assessing whether all pins of a component make proper contact with the board during soldering. If the pins are not coplanar, it could result in connection failures (*A Measurement Method that Solves Problems in Coplanarity Inspection*, n.d.). In this regard, a new feature called 'Coplanarity' was generated by using a lambda function that computes the difference between the maximum and minimum 'Height(um)' for each 'ComponentID'. This feature further enriches our dataset by providing insights into the uniformity of the solder paste height, which is critical for ensuring the quality and reliability of the solder joints.
- **Misalignment:** One of the main challenges in PCB assembly is the shifting or misalignment of components, which can arise due to issues such as incorrect solder application or excessive vibrations during the assembly process. Considering this, we extracted a new feature, 'Misalignment', to identify components with significant offsets in the X and Y directions. This new attribute was defined as any PCB components where the 'OffsetX(%)' and 'OffsetY(%)' are greater than 5. This threshold was chosen because an offset greater than 5% can indicate a significant deviation from the intended placement, potentially leading to connectivity issues or mechanical stress. The misalignment feature is a binary indicator where a value of 1, signifies that the component is misaligned (offset greater than 5% in either direction), or a value of 0, indicates proper alignment (*Design Mistakes That Cause PCB Assembly Errors*, n.d.). This feature is crucial for identifying and correcting potential issues in the placement process, ensuring higher precision and reliability.

- **Solder Characteristics:** Through careful analysis of the dataset, with particular focus on the 'Volume' feature and the associated AOI labels, two additional features were introduced to monitor the volume of solder paste named 'LeanSolder' and 'ExcessiveSolder'. The LeanSolder feature identifies instances where the '*Volume(%)*' is less than 80% , which can lead to insufficient solder joints, causing weak connections and potential failures. On the other hand, the ExcessiveSolder feature flags components where the '*Volume(%)*' exceeds 120%, which can result in solder bridging or other defects . These features are essential for ensuring optimal solder paste application, as deviations from the ideal volume can lead to defects and reliability issues in the final product.

- (5) **Temporal Features:** Temporal features capture changes in the production process over time, which can be crucial if defect occurrence is correlated with variations in production parameters. The inclusion of temporal features in defect detection models helps identify patterns and trends that may indicate underlying issues in the manufacturing process. This method is widely used in anomaly detection and demonstrated significant benefits in enhancing PCB images and improving defect recognition accuracy ([Putera & Ibrahim, 2010](#); [You, 2022](#)).

Temporal information in our dataset extracted from the features '*Date*' and '*Time*', incorporated as three additional features to capture temporal patterns in the inspection data. The extracted feature, "*Day_of_Week*", captures the day of the week when the inspection occurred, "*Hour_of_Day*" feature, captures the specific hour, and the "*Is_Weekend*" indicates whether the inspection took place on a weekend. These temporal features help in understanding patterns and trends in the inspection data and defective labels that could be related to operational shifts, work schedules, or other time-related factors.

It should be noted that, after extracting all important patterns and creating new features from the existing ones, all the original features used for feature engineering, including '*ComponentID*', '*FigureID*', '*PanelID*', '*PinNumber*', '*PadID*', '*Date*', '*Time*', '*PosX(mm)*', '*PosY(mm)*', '*PadType*', '*Volume(%)*', '*Height(um)*', '*Area(%)*', '*OffsetX(%)*', '*OffsetY(%)*', '*SizeX*', '*SizeY*', '*Volume(um3)*', '*Area(um2)*', '*Shape(um)*', '*Result*', and '*MachineID*' were excluded from the dataset. As a result, we ended up with **48** features in our dataset that were

used for model training. This step is crucial for several reasons. By extracting the essential information from the original features and creating new, more relevant features, we ensure that the dataset is optimized for the machine learning model. Removing the original features helps in simplifying the model, making it more interpretable and efficient. By focusing only on the newly engineered features, we can reduce the dimensionality of the dataset, improve the model's training time, and enhance its generalization capability.

4.2.4 Data Splitting

Moving forward to the final stage of data preprocessing, this task's approach to data splitting closely aligns with the methods outlined in the previous chapter (see Section 3.3.8), with a slight modification in the ratios used. Instead of the 80/20 split mentioned earlier, we implemented a 90/10 split for this phase. This adjustment in the splitting ratio ensures more extensive training with a focused validation process, aligning with our specific requirements for model refinement and testing.

4.3 Model Implementation

Building on the processes described in the previous chapter, once the data is prepared and preprocessed, the subsequent step is to select the models most likely to deliver optimal results. Given the recognized advantages and effectiveness of instance-based and tree-based models within an ensemble framework—particularly suited to handling highly imbalanced datasets—we have decided to continue employing these approaches. As with Chapter 3, we will first discuss the proposed models, exploring their suitability and the anticipated performance benefits and then conclude with an explanation of the library utilized for implementing these models, providing a comprehensive overview of our modeling infrastructure.

4.3.1 Proposed Models

This study employs advanced ensemble machine learning techniques to tackle the challenges posed by our dataset, particularly the issue of imbalanced classes. Ensemble models, sophisticated strategies in machine learning, are designed to enhance predictive performance by amalgamating the outputs of various individual models. The fundamental principle of these models is that a collection of weak learners can synergize to form a strong learner, thereby increasing the accuracy, robustness, and generalizability of the predictive model. One of the earliest and most effective ensemble methods is **bagging** (Bootstrap Aggregating), introduced by (Breiman, 1996). This technique involves training multiple models on different subsets of the training data and averaging their predictions to enhance stability and accuracy. Another influential technique, **boosting**, detailed initially by (Freund & Schapire, 1997) and further refined by (Friedman, 2001), sequentially trains models to correct the errors of their predecessors, then combines their outputs to form a robust predictive model.

Furthermore, in the training process, a hierarchical **stacking** method was employed, where models are layered sequentially (L1, L2, etc.), with each layer refining its predictions by integrating outputs from the previous layer's models. (Wolpert, 1992). This strategy allows the meta-model to potentially capture complex patterns in the interactions of the models' predictions. A notable implementation in our study is the '*DyStack*' feature, which dynamically adjusts the number of stacking layers, assessing whether adding layers enhances performance or leads to overfitting. This approach aims to balance model complexity with generalization capabilities, thereby optimizing overall performance. By effectively leveraging the strengths of individual models, the stacking approach significantly enhances the ensemble's accuracy and robustness.

Recent advancements in ensemble methods, especially those that integrate boosting techniques with tree-based models like LightGBM, have demonstrated considerable success across various applications, highlighting their robustness and versatility (Bokaba, Doorsamy, & Paul, 2022; Wei et al., 2022). In our work, the models chosen represent a strategic mix of KNN and LGBM learning algorithms, each augmented within an ensemble framework to optimize their predictive performance and robustness. As we have thoroughly discussed these models in a previous chapter (Section 3.4), here we briefly mention their role in the context of our ensemble strategy.

4.3.2 Tools and Libraries

In this study, we employed the **AutoGluon** library ([N. Erickson et al., 2020](#)), version 1.1.0, running on Python 3.10.12, to implement the proposed models. AutoGluon is an open-source machine-learning library designed to automate the creation and optimization of ML models. Traditionally, training and hyperparameter tuning processes can be labor intensive and time consuming. However, AutoGluon specializes in handling tabular data and excels at auto-tuning hyperparameters and autonomously selecting optimal models, significantly reducing the computational time and effort required. Particularly noteworthy is AutoGluon’s capability to apply advanced ensemble techniques, which have substantially enhanced our model’s performance and robustness. The results of these enhancements will be presented in Section [4.4.2](#).

4.4 Evaluation and Results

In this section, we initially focus on identifying the key metric that is essential for accurately evaluating and optimizing our machine learning models. Choosing the correct metric is paramount as it directly influences our perception of the models’ effectiveness and efficiency. Upon identifying this metric, we will conduct a thorough analysis of the model training and outcomes from the proposed models, assessing their efficacy to formulate well-substantiated conclusions. Additionally, an in-depth analysis of feature importance will be performed to provide crucial insights into which engineered attributes most significantly impact our model’s predictions. This analysis aims to highlight the predictive power and relevance of individual features, further informing the refinement and potential enhancements of our modeling approach. Lastly, this section ends with a comparative analysis of the different models employed, discussing their strengths and weaknesses in the context of the study.

4.4.1 Metrics

As emphasized in Section [4.2.1](#), our objective is to predict whether components with confirmed defects are categorized as "FalseScrap" or "NotPossibleToRepair." This prediction task involves using the RepairLabel. For predicting these class labels, selecting appropriate metrics is crucial

for accurately interpreting model performance, especially in scenarios involving imbalanced class distributions. In this regard, we have adopted the F1 macro metric to assess our models. As detailed in Formula 20, this metric computes the harmonic mean of precision and recall for each class independently before averaging these values:

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^N \left(\frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \right) \quad (20)$$

where N is the number of classes, and Precision_i and Recall_i are the precision and recall for the i -th class, respectively.

This method ensures that all classes are treated with equal importance, which is particularly beneficial in our dataset, characterized by uneven class distributions. Utilizing the F1 macro metric enables a thorough evaluation of the model’s performance across all classes, highlighting its ability to handle infrequent classes as effectively as the more common ones. In the following section, we present our training process and results using this macro-averaging method to calculate the key metric, providing insights into the model’s overall accuracy and robustness across varied class distributions.

4.4.2 Model Training and Results

As previously discussed, the F1 macro score was selected as the primary metric for evaluating our predictive models due to its inclusion of both precision and recall—key components that are crucial given the aims of our study and the nature of our data. In this regard, ensemble models were configured in AutoGluon to optimize this metric, highlighting its appropriateness for datasets where class imbalance is present. Our modeling approach involved two stack levels, Level 1 (L1), where individual models are directly trained on the dataset and Level 2 (L2), where more advanced models are developed utilizing the predictions from L1 models as inputs. This hierarchical stacking is designed to enhance predictive accuracy by integrating and refining outputs across model levels.

As we mentioned in Section 4.3.1, during the training phase, we implemented a bagging framework as part of our ensemble strategy. In this framework, each primary model was responsible for

fitting eight subsidiary models, known as child models. These child models were trained using subsets of the full dataset, selected randomly with replacement. This method of sampling ensures that each model experiences a variety of data scenarios, thereby incorporating a wide range of data characteristics into the training process. This diversity is crucial for the models to effectively capture and learn from the complex patterns present in the dataset, enhancing the model’s robustness and generalizing their predictive capabilities across different data conditions. Combining bagging with stacking introduces a powerful synergy in the modeling process. While stacking layers are used to refine and enhance predictions vertically through the model layers, bagging spreads the training horizontally across multiple versions of the same model level. This dual approach significantly strengthens the model’s ability to generalize well across different data scenarios and conditions.

The results, detailed in Table 4.1, indicate that the KNN model at L1 achieved notable success. Specifically, this model employing a uniform weighting strategy where all neighbors contribute equally, achieved an F1 macro score of 0.811 on the test set. This performance suggests that a uniform approach to neighbor weighting can be more effective than distance-based weighting, particularly in complex classification scenarios. This model demonstrated consistent performance across both validation and test sets, indicating superior generalization compared to the distance-weighted variant, which, while performing exceptionally well on the validation set (F1 macro score of 0.892), showed a 9% drop in performance on the test set. In addition to these advantages, this model showed the lowest prediction time, making it highly suitable for real-time applications.

Table 4.1: Model Performance Results

Model	Score Test	Score Val	Pred Time Test (s)	Pred Time Val (s)
KNeighborsUnif_BAG_L1	0.811	0.819	0.008	0.038
KNeighborsDist_BAG_L1	0.808	0.892	0.004	0.098
LightGBMXT_BAG_L1	0.707	0.905	1.950	0.020
LightGBMXT_BAG_L2	0.701	0.957	2.180	0.112

Switching our focus to the LightGBM models enhanced with Extra Trees (LightGBMXT), this

model at stack level 1, showed robust pattern recognition and generalization on the validation set with an F1 macro score of 0.905. This performance was further improved at L2, where it exhibited a 5% increase in the F1 macro score on the validation set. However, the performance of these models on the test set was significantly lower, scored 0.707 and 0.701 respectively. This result highlights the challenge of capturing complex patterns present in the test data. These discrepancies between validation and test performances underscore the necessity of robust cross-validation techniques and careful model tuning to ensure that models not only perform well in controlled experimental conditions but also maintain their effectiveness in practical, real-world settings. The study's findings emphasize the importance of choosing appropriate model configurations and validation strategies to achieve reliable and generalizable predictive performance.

On the other hand, the analysis of stacking levels indicated that increasing the complexity through stacking did not consistently yield improvements in performance across the test data, as seen in the table. This suggests that while stacking can enhance model robustness and generalization to validation data, its advantages might not translate as effectively to diverse real-world conditions as represented on the test dataset. Therefore, a single stack level appears to provide an optimal balance of performance and computational efficiency.

4.4.3 Feature importance Analysis

In this section, we employed the Permutation method to evaluate the importance of features in predicting repair labels on PCB circuit boards. This technique assesses feature importance by evaluating the decrease in a model's performance when the values of each feature are shuffled randomly. The shuffling is repeated 5 times to ensure the reliability of the importance estimate, addressing random fluctuations by averaging the effects. This analysis highlights several key features that significantly influence our model's performance. As shown in Figure 4.2, a bar chart format visually compares the top ten features based on their importance.

As demonstrated, the volume-related features, particularly those extracted during feature aggregation (see Section 3) emerge as the most critical, with 'Min.Volume.Component' alone accounting for nearly 25% of the importance in our model. Furthermore, soldering status features such as 'AOILabel_Translated', 'AOILabel_UnSoldered', and 'AOILabel_Soldered' also show considerable

influence, suggesting that these features directly impact our model prediction. This feature importance profile underscores the value of focusing on soldering volume and quality in predicting repair needs, which can guide more targeted quality control measures in PCB manufacturing processes.

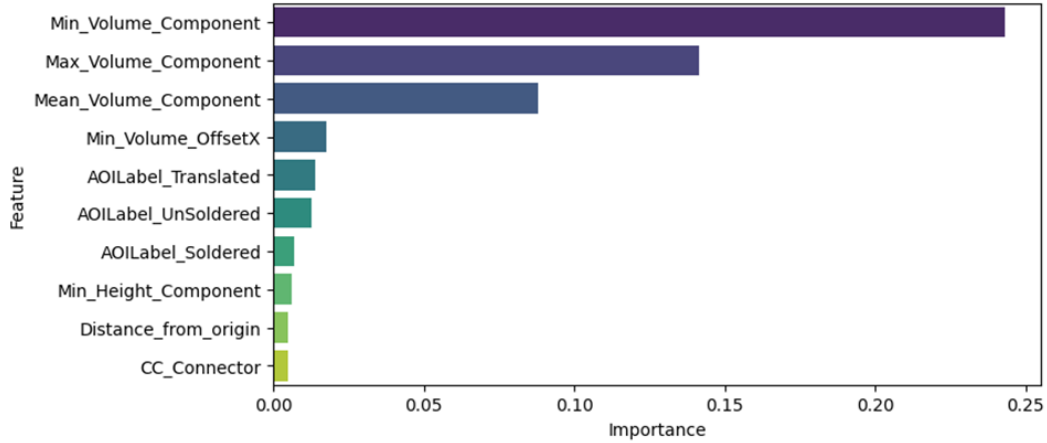


Figure 4.2: Feature Importance Analysis

4.4.4 Comparative Review of Our Methods Against Current Works

As we navigate the complexities of ensemble ML approaches in PCB manufacturing, it becomes crucial to understand how our methods compare to existing research. This section serves as a comparative analysis, designed to show how our work fits into the bigger picture of similar research. Table 4.2 provides a comparative summary with four related studies. Even though the best training performance in other studies such as Case 1 and Case 2 achieved an F1-score of 0.90, their lower test set results indicate that their models might not be properly trained.

In stark contrast, our approach not only maintains consistency between training and test performances but also excels in test environments with an F1 score of 0.81. This indicates a robust model that generalizes well beyond the training data—a critical attribute for practical applications in the PCB manufacturing sector. Our method not only avoids the pitfalls of overfitting observed in the comparative cases but also improves upon the highest test scores among the referenced studies. This improvement is significant, underscoring the effectiveness of our ensemble models and strategic feature engineering techniques. These results highlight our method’s superiority in applying

Table 4.2: F1-score Comparison for RepairLabel Prediction

Comparison Cases	Train	Test
Our Method	0.81	0.81
Case 1 (Gaffet et al., 2022)	0.90	0.77
Case 2 (Gore et al., 2022)	0.90	0.78
Case 3 (Tang et al., 2022)	0.83	0.71
Case 4 (Schmidt et al., 2022)	0.87	0.70

learned patterns to new, unseen data, which is essential for deployment in real-world settings where reliability and accuracy are critical.

4.5 Summary of The Chapter

This chapter explored enhancements in quality control within PCB manufacturing by automating the assignment of repair labels. Building on the earlier study that assessed PCB condition following component replacement through "OperatorLabel", this segment concentrates on automating the "RepairLabel" process, which is currently performed manually. We introduced machine learning ensemble models designed to predict these labels with high precision, potentially eliminating the need for human intervention. Additionally, the chapter detailed the dataset employed, emphasizing specific preprocessing actions and feature engineering techniques used.

The findings from this chapter affirm the viability of ML models such as KNN and LGBM, to substantially improve the quality control process in PCB manufacturing through automation. The demonstrated success of uniform weighting strategies in KNN models, coupled with the prudent use of bagging techniques, offers a promising direction for future research and practical implementations in the industry. Overall, this chapter not only highlights the technological advancements in automated systems but also sets a benchmark for future innovations aimed at optimizing manufacturing processes.

Chapter 5

Summary and Future Research Directions

This chapter serves as the conclusion of the thesis, summarizing its key contributions. Additionally, it will offer insights into potential future research directions in this field, which are discussed towards the end of the chapter.

5.1 Summary of Thesis Contributions

In this research, we have made several contributions aimed at enhancing the quality and stability of PCB circuit board production lines. We achieved this by employing a diverse ensemble of machine learning models, ranging from Instance-Based models to Tree-Based and Boosting models. Furthermore, we utilized the TabNet model, specifically designed for tabular datasets, to thoroughly analyze the outcomes.

The exploration began with an initial focus on quality control post-component placement. We highlighted the comparative advantages of ML/DL models over human operators in AOI defect detection, noting superior speed, consistency, and reduced error rates of these models, despite the value added by human insight. In this phase, we addressed several critical challenges associated with the imbalanced nature of our dataset, particularly in the context of PCB circuit board production. Our primary focus was on mitigating these challenges through a comprehensive data augmentation

strategy. We employed two key techniques, a simpler method like SMOTE and a more sophisticated GAN-based model such as CTGAN. After overcoming the challenges associated with generating synthetic data, we introduced a novel perspective on integrating this data into the original dataset. By systematically adding synthetic data generated by both models in varying volumes, we were able to assess its impact on model performance and optimization. The strategic integration of synthetic data, particularly with a 40% to 60% inclusion, led to significant improvements in crucial metrics like the F1 score, with the KNN Classifier and TabNet model notably excelling in this regard.

The second chapter expanded the narrative to include automation in the repair label assignment process, traditionally a manual task. In this phase of our research, we proposed a new data preprocessing approach by leveraging advanced feature engineering methods. Through techniques such as feature extraction, feature transformation, and feature aggregation, we significantly enhanced the model's ability to learn complex patterns within the dataset. Ultimately, by employing ensemble methods, we developed a robust and high-performing model, which contributes substantially to improving the quality control processes in PCB manufacturing. A consistent observation was the comparative advantage of the KNN Classifier across various applications, affirming its robustness and versatility as a machine learning model.

5.2 Future Research

The current study opens several avenues for future work in the domain of anomaly detection in PCB manufacturing. In future research, further exploration into other GAN-based methodologies presents a promising avenue for enhancing anomaly detection in PCB manufacturing. By developing and integrating novel GAN architectures, researchers can tailor these models to better handle the specific challenges presented by the SPI and AOI datasets used in both Operator Label and Repair Label phases. This approach would involve rigorous evaluation to compare the effectiveness of different GAN configurations in synthesizing realistic, yet diverse, training samples that improve model robustness.

Additionally, leveraging hybrid ML and DL models could significantly boost performance. By combining the strengths of traditional machine learning techniques with advanced deep learning

frameworks, these hybrid models could offer more nuanced feature extraction capabilities and improved decision-making processes, thereby increasing accuracy and reducing false positives in anomaly detection tasks.

Another promising avenue could be Investigating multi-task learning approaches that could allow simultaneous detection of multiple defect types or tasks, improving overall model performance. Techniques such as transfer learning could be employed to leverage knowledge from related tasks, enhancing the model's adaptability.

Appendix A

Evaluation of Imbalance Techniques Using Various Synthetic Data Volumes for Operator Label Prediction

Table A.1: Cat Boost Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1.Score
-	858	Valid	0.987±0.001	0.947±0.014	0.931±0.036	0.727±0.022	0.816±0.024
		Test	0.963	0.817	0.770	0.309	0.442
0.2	4310	Valid	0.983±0.002	0.990±0.004	0.981±0.008	0.916±0.016	0.947±0.006
		Test	0.963	0.780	0.710	0.345	0.464
0.4	8620	Valid	0.982±0.002	0.995±0.001	0.987±0.002	0.950±0.006	0.968±0.004
		Test	0.963	0.806	0.696	0.373	0.486
0.6	12930	Valid	0.981±0.003	0.996±0.001	0.991±0.003	0.960±0.008	0.975±0.005
		Test	0.963	0.804	0.689	0.371	0.482
0.8	17240	Valid	0.981±0.002	0.997±0.0007	0.990±0.003	0.968±0.005	0.979±0.003
		Test	0.960	0.805	0.626	0.375	0.469
minority	21551	Valid	0.981±0.002	0.997±0.0004	0.991±0.002	0.972±0.004	0.980±0.002
		Test	0.960	0.804	0.623	0.394	0.482

Table A.2: Gradient Boosting Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.986±0.002	0.943±0.014	0.919±0.046	0.699±0.035	0.794±0.032
		Test	0.959	0.864	0.626	0.294	0.400
0.2	4310	Valid	0.966±0.004	0.969±0.009	0.949±0.013	0.840±0.023	0.891±0.014
		Test	0.953	0.833	0.496	0.334	0.399
0.4	8620	Valid	0.954±0.004	0.977±0.002	0.964±0.009	0.871±0.009	0.915±0.007
		Test	0.951	0.847	0.475	0.389	0.428
0.6	12930	Valid	0.949±0.004	0.980±0.003	0.969±0.006	0.893±0.010	0.930±0.006
		Test	0.953	0.847	0.497	0.415	0.452
0.8	17240	Valid	0.946±0.003	0.980±0.003	0.974±0.004	0.902±0.008	0.937±0.004
		Test	0.951	0.846	0.473	0.428	0.449
Minority	21551	Valid	0.943±0.003	0.981±0.001	0.974±0.003	0.909±0.005	0.941±0.001
		Test	0.947	0.842	0.440	0.452	0.446

Table A.3: Extra Trees Performance Metrics at Different Levels of SMOTE Oversampling

SMOTE(k=10) Sampling Strategy	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
-	858	Valid	0.986±0.002	0.934±0.016	0.911±0.044	0.707±0.034	0.795±0.034
		Test	0.961	0.824	0.669	0.316	0.429
0.2	4310	Valid	0.985±0.002	0.991±0.003	0.973±0.006	0.936±0.015	0.954±0.007
		Test	0.960	0.851	0.634	0.369	0.466
0.4	8620	Valid	0.985±0.002	0.995±0.001	0.982±0.003	0.965±0.006	0.973±0.004
		Test	0.960	0.844	0.623	0.400	0.487
0.6	12930	Valid	0.985±0.002	0.997±0.0009	0.985±0.003	0.975±0.005	0.980±0.003
		Test	0.960	0.840	0.605	0.417	0.494
0.8	17240	Valid	0.985±0.001	0.998±0.0006	0.987±0.003	0.974±0.004	0.984±0.001
		Test	0.958	0.844	0.575	0.417	0.484
minority	21551	Valid	0.986±0.002	0.998±0.0005	0.987±0.002	0.984±0.003	0.986±0.002
		Test	0.957	0.839	0.561	0.422	0.481

Table A.4: Random Forest Performance Metrics at Different Levels of CT-GAN Synthetic Data

Imbalance Technique	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.986±0.001	0.932±0.021	0.943±0.031	0.694±0.026	0.799±0.023
		Test	0.961	0.844	0.716	0.294	0.410
	4310	Valid	0.988±0.002	0.987±0.004	0.989±0.006	0.940±0.011	0.964±0.007
		Test	0.961	0.841	0.689	0.327	0.444
	8620	Valid	0.989±0.001	0.993±0.001	0.994±0.003	0.969±0.005	0.981±0.003
		Test	0.962	0.850	0.683	0.364	0.475
CT-GAN	12930	Valid	0.990±0.001	0.996±0.001	0.996±0.002	0.979±0.004	0.987±0.002
		Test	0.963	0.864	0.706	0.375	0.49
	17240	Valid	0.991±0.001	0.996±0.0008	0.996±0.001	0.985±0.002	0.990±0.001
		Test	0.963	0.863	0.693	0.373	0.485
	21551	Valid	0.992±0.001	0.997±0.0007	0.997±0.001	0.987±0.002	0.992±0.001
		Test	0.962	0.858	0.690	0.362	0.475

Table A.5: Decision Tree Performance Metrics at Different Levels of CT-GAN Synthetic Data

Imbalance Technique	Number of Defectives	Data	Accuracy	AUC	Precision	Recall	F1_Score
None	858	Valid	0.977±0.003	0.857±0.022	0.691±0.049	0.727±0.044	0.708±0.044
		Test	0.959	0.676	0.595	0.364	0.452
	4310	Valid	0.987±0.002	0.986±0.005	0.982±0.006	0.940±0.011	0.960±0.006
		Test	0.954	0.658	0.520	0.331	0.405
	8620	Valid	0.981±0.002	0.976±0.003	0.968±0.006	0.965±0.004	0.966±0.005
		Test	0.956	0.675	0.555	0.364	0.44
CT-GAN	12930	Valid	0.982±0.002	0.981±0.002	0.977±0.004	0.977±0.004	0.977±0.003
		Test	0.954	0.658	0.513	0.331	0.403
	17240	Valid	0.983±0.002	0.983±0.002	0.982±0.004	0.988±0.002	0.981±0.002
		Test	0.953	0.662	0.498	0.340	0.404
	21551	Valid	0.984±0.002	0.984±0.002	0.985±0.002	0.985±0.003	0.983±0.002
		Test	0.9529	0.6617	0.490	0.340	0.402

References

- Ackerman, S., Kour, G., & Farchi, E. (2023). Characterizing how 'distributional' nlp corpora distance metrics are. *arXiv preprint arXiv:2310.14829*.
- Adamson, T., Mahmud, M., Wu, Y., Lin, N., Diao, F., Zhao, Y., & Balda, J. C. (2020). An 800-v high-density traction inverter—electro-thermal characterization and low-inductance pcb bussing design. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 10(3), 3013–3023.
- Adibhatla, V. A., Chih, H.-C., Hsu, C.-C., Cheng, J., Abbod, M. F., & Shieh, J.-S. (2021). Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once.
- Ahsan, M. M., Mahmud, M. P., Saha, P. K., Gupta, K. D., & Siddique, Z. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, 9(3), 52.
- Albee, A. J. (2013). The evolution of ict: Pcb technologies, test philosophies, and manufacturing business models are driving in-circuit test evolution and innovations. In *Ipc apex expo conference and exhibition* (Vol. 1, pp. 381–401).
- Ana-Maria, B., Georgiana, H. C., & Ioan, L. (2018). Stand for the functional testing automation of the electronic modules. In *2018 international symposium on electronics and telecommunications (isetc)* (pp. 1–4).
- Anoop, K., Sarath, N., & Kumar, V. (2015). A review of pcb defect detection using image processing. *Intern. Journal of Engineering and Innovative Technology (IJEIT)*, 4(11), 188–192.
- Arik, S. Ö., & Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. In *Proceedings of*

- the aaai conference on artificial intelligence* (Vol. 35, pp. 6679–6687).
- Bai, L., & Kalaj, D. (2021). Approximation of kolmogorov–smirnov test statistic. *Stochastics*, 93(7), 993–1027.
- Bokaba, T., Doorsamy, W., & Paul, B. S. (2022). A comparative study of ensemble models for predicting road traffic congestion. *Applied Sciences*, 12(3), 1337.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24, 123–140.
- Breiman, L. (2017). *Classification and regression trees*. Routledge.
- Bullejos, M., Cabezas, D., Martín-Martín, M., & Alcalá, F. J. (2022). A k-nearest neighbors algorithm in python for visualizing the 3d stratigraphic architecture of the llobregat river delta in ne spain. *Journal of Marine Science and Engineering*, 10(7), 986.
- Cao, X. H., Stojkovic, I., & Obradovic, Z. (2016). A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC bioinformatics*, 17, 1–10.
- Chakraborty, M., Kettle, J., & Dahiya, R. (2022). Electronic waste reduction through devices and printed circuit boards designed for circularity. *IEEE Journal on Flexible Electronics*, 1(1), 4–23.
- Chaudhary, V., Dave, I. R., & Upla, K. P. (2017). Automatic visual inspection of printed circuit board for defect detection and classification. In *2017 international conference on wireless communications, signal processing and networking (wisnet)* (pp. 732–737).
- Chauhan, A. P. S., & Bhardwaj, S. C. (2011). Detection of bare pcb defects by image subtraction method using machine vision. In *Proceedings of the world congress on engineering* (Vol. 2, pp. 6–8).
- Chen, I.-C., Hwang, R.-C., & Huang, H.-C. (2023). Pcb defect detection based on deep learning algorithm. *Processes*, 11(3), 775.
- Chen, J., Zhang, Z., & Wu, F. (2021). A data-driven method for enhancing the image-based automatic inspection of ic wire bonding defects. *International journal of production research*, 59(16), 4779–4793.
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

- Chen, X., Tao, L., Shang, J., & Liu, D. (2022). Application research of ultrasonic testing in microwave multilayer pcb. In *Proceedings of the eighth asia international symposium on mechatronics* (pp. 1935–1943).
- Corander, J., Remes, U., & Koski, T. (2021). On the jensen-shannon divergence and the variation distance for categorical probability distributions. *Kybernetika*, 57(6), 879–907.
- Cui, H., & Anderson, C. G. (2016). Literature review of hydrometallurgical recycling of printed circuit boards (pcbs). *J. Adv. Chem. Eng*, 6(1), 142–153.
- Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. *Ensemble machine learning: Methods and applications*, 157–175.
- Dai, W., Mujeeb, A., Erdt, M., & Sourin, A. (2018). Towards automatic optical inspection of soldering defects. In *2018 international conference on cyberworlds (cw)* (pp. 375–382).
- Dai, W., Mujeeb, A., Erdt, M., & Sourin, A. (2020). Soldering defect detection in automatic optical inspection. *Advanced Engineering Informatics*, 43, 101004.
- Dervišević, I., Minić, D., Kamberović, Ž., Čosović, V., & Ristić, M. (2013). Characterization of pcbs from computers and mobile phones, and the proposal of newly developed materials for substitution of gold, lead and arsenic. *Environmental Science and Pollution Research*, 20, 4278–4292.
- Design mistakes that cause pcb assembly errors.* (n.d.). Retrieved from <https://www.protoexpress.com/blog/design-mistakes-that-cause-pcb-assembly-errors> (accessed September 1, 2024)
- Eom, G., & Byeon, H. (2023). Searching for optimal oversampling to process imbalanced data: Generative adversarial networks and synthetic minority over-sampling technique. *Mathematics*, 11(16), 3605.
- Erickson, B. J., & Kitamura, F. (2021). *Magician’s corner: 9. performance metrics for machine learning models* (Vol. 3) (No. 3). Radiological Society of North America.
- Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020). Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*.
- Farkas, C., Géczy, A., Kovács, R., & Bonyár, A. (2022). Biodegradable and nanocomposite substrates: new prospects for sustainable electronics packaging. *IEEE EPS eNews*, 9, 1–9.

- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1), 119–139.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.
- Gaber, L., Hussein, A. I., & Moness, M. (2021). Fault detection based on deep learning for digital vlsi circuits. *Procedia Computer Science*, 194, 122–131.
- Gaffet, A., Roa, N. B., Ribot, P., Chanthery, E., & Merle, C. (2022). A hierarchical xgboost early detection method for quality and productivity improvement of electronics manufacturing systems. In *Phm society european conference* (Vol. 7).
- Galetto, M., Verna, E., Genta, G., & Franceschini, F. (2020). Uncertainty evaluation in the prediction of defects and costs for quality inspection planning in low-volume productions. *The International Journal of Advanced Manufacturing Technology*, 108, 3793–3805.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63, 3–42.
- Gore, P., Minami, T., Kundu, P., Lee, J., et al. (2022). A novel methodology for health assessment in printed circuit boards. In *Phm society european conference* (Vol. 7, pp. 556–562).
- Houdek, C., & Design, C. (2016). Inspection and testing methods for pcbs: An overview. *Engineer/OwnerCaltronics Design & Assembly*.
- Huang, R., Gu, J., Sun, X., Hou, Y., & Uddin, S. (2019). A rapid recognition method for electronic components based on the improved yolo-v3 network. *Electronics*, 8(8), 825.
- Jemai, J., & Zarrad, A. (2023). Feature selection engineering for credit risk assessment in retail banking. *Information*, 14(3), 200.
- Jin, J., Feng, W., Lei, Q., Gui, G., Li, X., Deng, Z., & Wang, W. (2021). Defect detection of printed circuit boards using efficientdet. In *2021 ieee 6th international conference on signal and image processing (icsip)* (pp. 287–293).
- Jun, H., & Jung, I. Y. (2023). Enhancement of product-inspection accuracy using convolutional neural network and laplacian filter to automate industrial manufacturing processes. *Electronics*, 12(18), 3795.

- Jurj, S. L., Rotar, R., Opritoiu, F., & Vladutiu, M. (2020). Affordable flying probe-inspired in-circuit-tester for printed circuit boards evaluation with application in test engineering education. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (IEEEIC/ICPS Europe)* (pp. 1–6).
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Khalilian, S., Hallaj, Y., Balouchestani, A., Karshenas, H., & Mohammadi, A. (2020). Pcb defect detection using denoising convolutional autoencoders. In *2020 International Conference on Machine Vision and Image Processing (MVIP)* (pp. 1–5).
- Kim, J., Ko, J., Choi, H., & Kim, H. (2021). Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors*, 21(15), 4968.
- Laria, H., Wang, Y., van de Weijer, J., & Raducanu, B. (2022). Transferring unconditional to conditional gans with hyper-modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3840–3849).
- Li, Y.-T., Kuo, P., & Guo, J.-I. (2020). Automatic industry pcb board dip process defect detection with deep ensemble method. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)* (pp. 453–459).
- Lin, C.-H., Wang, S.-H., & Lin, C.-J. (2019). Using convolutional neural networks for character verification on integrated circuit components of printed circuit boards. *Applied Intelligence*, 49(11), 4022–4032.
- Ling, Q., & Isa, N. A. M. (2023). Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey. *IEEE Access*.
- Liu, X., & Hsieh, C.-J. (2019). Rob-gan: Generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11234–11243).
- Luque-Rodriguez, M., Molina-Baena, J., Jimenez-Vilchez, A., & Arauzo-Azofra, A. (2022). Initialization of feature selection search for classification. *Journal of Artificial Intelligence Research*, 75, 953–983.

- Mashohor, S., Evans, J. R., & Erdogan, A. T. (2006). Automatic hybrid genetic algorithm based printed circuit board inspection. In *First nasa/esa conference on adaptive hardware and systems (ahs'06)* (pp. 390–400).
- A measurement method that solves problems in coplanarity inspection. (n.d.). Retrieved from <https://www.keyence.com/ss/products/microscope/measurement-solutions/coplanarity.jsp> ([Online])
- Mendikowski, M., & Hartwig, M. (2022). Creating customers that never existed: Synthesis of e-commerce data using ctgan. In *18th international conference on machine learning and data mining (mldm-22). new york, us: Ibai publishing* (pp. 91–105).
- Mescheder, L., Geiger, A., & Nowozin, S. (2018). Which training methods for gans do actually converge? In *International conference on machine learning* (pp. 3481–3490).
- Mirzaei, M. (2023). *Automating fault detection and quality control in pcbs: A machine learning approach to handle imbalanced data* (Unpublished doctoral dissertation). Concordia University.
- Neubauer, C. (1997). Intelligent x-ray inspection for quality control of solder joints. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C*, 20(2), 111–120.
- Neubauer, C., & Hanke, R. (1993). Improving x-ray inspection of printed circuit boards by integration of neural network classifiers. In *Proceedings of 15th ieee/chmt international electronic manufacturing technology symposium* (pp. 14–18).
- Ng, A., et al. (2011). Sparse autoencoder. *CS294A Lecture notes*, 72(2011), 1–19.
- Nguyen, V.-T., & Bui, H.-A. (2022). A real-time defect detection in printed circuit boards applying deep learning. *EUREKA: Physics and Engineering*, (2), 143–153.
- Nkikabahizi, C., Cheruiyot, W., & Kibe, A. (2022). Chaining zscore and feature scaling methods to improve neural networks for classification. *Applied Soft Computing*, 123, 108908.
- Ong, A. T., Mustapha, A., Ibrahim, Z. B., Ramli, S., & Eong, B. C. (2015). Real-time automatic inspection system for the classification of pcb flux defects. *American Journal of Engineering and Applied Sciences*, 8(4), 504.
- Park, J., Kwon, S., & Jeong, S.-P. (2023). A study on improving turnover intention forecasting by solving imbalanced data problems: focusing on smote and generative adversarial networks.

- Journal of Big Data*, 10(1), 36.
- Parlak, I. E., & Emel, E. (2023). Deep learning-based detection of aluminum casting defects and their types. *Engineering Applications of Artificial Intelligence*, 118, 105636.
- Patki, N., Wedge, R., & Veeramachaneni, K. (2016, Oct). The synthetic data vault. In *Ieee international conference on data science and advanced analytics (dsaa)* (p. 399-410). doi: 10.1109/DSAA.2016.49
- Paul, S. D., & Bhunia, S. (2021). Silverin: Systematic integrity verification of printed circuit board using jtag infrastructure. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 17(3), 1–28.
- Perdigones, F., & Quero, J. M. (2022). Printed circuit boards: The layers' functions for electronic and biomedical engineering. *Micromachines*, 13(3), 460.
- PHM.Society. (2022). *data challenge: 7th european conference of the prognostics and health management society 2022*. Retrieved 18.06.2022, from <https://phm-europe.org/data-challenge>
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.
- Putera, S. I., & Ibrahim, Z. (2010). Printed circuit board defect detection using mathematical morphology and matlab image processing tools. In *2010 2nd international conference on education technology and computer* (Vol. 5, pp. V5–359).
- Pycaret — pycaret 2.3.5 documentation*. (n.d.). Retrieved from <https://pycaret.readthedocs.io/en/stable/> (Accessed: 2022-11-10)
- Pytorch*. (n.d.). Retrieved from <https://pytorch.org/> (Accessed: 2022-11-10)
- Regol, F., Kroon, A., & Coates, M. (2023). Evaluation of categorical generative models-bridging the gap between real and synthetic data. In *Icassp 2023-2023 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 1–5).
- Reitermanova, Z., et al. (2010). Data splitting. In *Wds* (Vol. 10, pp. 31–36).
- Sankar, V. U., Lakshmi, G., & Sankar, Y. S. (2022). A review of various defects in pcb. *Journal of Electronic Testing*, 38(5), 481–491.

- Sarawade, A. A., & Charniya, N. N. (2019). Detection of faulty integrated circuits in pcb with thermal image processing. In *2019 international conference on nascent technologies in engineering (icnte)* (pp. 1–6).
- Schmidt, I., Dingeldein, L., Hünemohr, D., Simon, H., & Weigert, M. (2022). Application of machine learning methods to predict the quality of electric circuit boards of a production line. In *Phm society european conference* (Vol. 7, pp. 550–555).
- Shah, C., Du, Q., & Xu, Y. (2022). Enhanced tabnet: Attentive interpretable tabular learning for hyperspectral image classification. *Remote Sensing*, *14*(3), 716.
- Shashidhara, H., Yellampalii, S., & Goudanavar, V. (2014). Board level jtag/boundary scan test solution. In *International conference on circuits, communication, control and computing* (pp. 73–76).
- Silva, L. H. d. S., Azevedo, G. O. d. A., Fernandes, B. J., Bezerra, B. L., Lima, E. B., & Oliveira, S. C. (2019). Automatic optical inspection for defective pcb detection using transfer learning. In *2019 ieee latin american conference on computational intelligence (la-cci)* (pp. 1–6).
- Singh, K., Kharche, S., Chauhan, A., & Salvi, P. (2024). Pcb defect detection methods: A review of existing methods and potential enhancements. *Journal of Engineering Science & Technology Review*, *17*(1).
- Sundaram, S., & Zeid, A. (2023). Artificial intelligence-based smart quality inspection for manufacturing. *Micromachines*, *14*(3), 570.
- Ta, V.-C., Hoang, T.-L., Doan, N.-S., Nguyen, V.-T., Nguyen Dieu, N., Pham, T. T. T., & Nguyen Dang, N. (2023). Tabnet efficiency for facies classification and learning feature embedding from well log data. *Petroleum Science and Technology*, 1–16.
- Tang, H., Tian, Y., Dai, J., Wang, Y., Cong, J., Liu, Q., ... Fu, Y. (2022). Prediction of production line status for printed circuit boards. In *Phm society european conference* (Vol. 7, pp. 563–570).
- Taunk, K., De, S., Verma, S., & Swetapadma, A. (2019). A brief review of nearest neighbor algorithm for learning and classification. In *2019 international conference on intelligent computing and control systems (iccs)* (pp. 1255–1260).
- Thiyam, B., & Dey, S. (2024). Ciir: an approach to handle class imbalance using a novel feature

- selection technique. *Knowledge and Information Systems*, 1–34.
- Tsai, D.-M., & Huang, C.-K. (2018). Defect detection in electronic surfaces using template-based fourier image reconstruction. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 9(1), 163–172.
- Ural, D. I., & Sezen, A. (2024). Research on pcb defect detection using artificial intelligence: a systematic mapping study. *Evolutionary Intelligence*, 1–11.
- Vafeiadis, T., Dimitriou, N., Ioannidis, D., Wotherspoon, T., Tinker, G., & Tzovaras, D. (2018). A framework for inspection of dies attachment on pcb utilizing machine learning techniques. *Journal of Management Analytics*, 5(2), 81–94.
- Verma, S., Arora, V., & Perumal, T. (2023). Art generation ai model for low-end devices. In *2023 international conference on advances in computation, communication and information technology (icaiccit)* (pp. 30–35).
- Verna, E., Genta, G., Galetto, M., & Franceschini, F. (2023). Zero defect manufacturing: a self-adaptive defect prediction model based on assembly complexity. *International Journal of Computer Integrated Manufacturing*, 36(1), 155–168.
- Verna, E., Puttero, S., Genta, G., & Galetto, M. (2023). Toward a concept of digital twin for monitoring assembly and disassembly processes. *Quality Engineering*, 1–18.
- Wan, J., Chen, H., Yuan, Z., Li, T., Yang, X., & Sang, B. (2021). A novel hybrid feature selection method considering feature interaction in neighborhood rough set. *Knowledge-Based Systems*, 227, 107167.
- Wang, F., Yue, Z., Liu, J., Qi, H., Sun, W., Chen, M., ... Yue, H. (2022). Quantitative imaging of printed circuit board (pcb) delamination defects using laser-induced ultrasound scanning imaging. *Journal of Applied Physics*, 131(5).
- Wang, S., Luo, H., Huang, S., Li, Q., Liu, L., Su, G., & Liu, M. (2023). Counterfactual-based minority oversampling for imbalanced classification. *Engineering Applications of Artificial Intelligence*, 122, 106024.
- Wei, A., Yu, K., Dai, F., Gu, F., Zhang, W., & Liu, Y. (2022). Application of tree-based ensemble models to landslide susceptibility mapping: A comparative study. *Sustainability*, 14(10), 6330.

- Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2), 241–259.
- Worden, J. C. (2024). Elevating excellence in high-reliability electronics: The personal and lasting legacy of clean pcbs. In *2024 pan pacific strategic electronics symposium (pan pacific)* (pp. 1–2).
- Wu, C. (2020). Research on design and application of brand vision inspection and sorting system based on image processing. In *2020 2nd international conference on machine learning, big data and business intelligence (mlbdbi)* (pp. 568–571).
- Wu, C., Awasthi, A. K., Qin, W., Liu, W., & Yang, C. (2022). Recycling value materials from waste pcbs focus on electronic components: technologies, obstruction and prospects. *Journal of Environmental Chemical Engineering*, 10(5), 108516.
- Wu, H., Lei, R., & Peng, Y. (2022). Pcbnet: A lightweight convolutional neural network for defect inspection in surface mount technology. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–14.
- Xu, K., Li, C., Zhu, J., & Zhang, B. (2019). Understanding and stabilizing gans’ training dynamics with control theory. *arXiv preprint arXiv:1909.13188*.
- Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32.
- Yang, J., Li, S., Wang, Z., Dong, H., Wang, J., & Tang, S. (2020). Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges. *Materials*, 13(24), 5755.
- Ye, D., Wang, X., & Chen, X. (2023). Lightweight generative joint source-channel coding for semantic image transmission with compressed conditional gans. In *2023 ieee/cic international conference on communications in china (iccc workshops)* (pp. 1–6).
- Yeom, T., Gu, C., & Lee, M. (2024). Dudgan: Improving class-conditional gans via dual-diffusion. *IEEE Access*.
- You, S. (2022). Pcb defect detection based on generative adversarial network. In *2022 2nd international conference on consumer electronics and computer engineering (iccece)* (pp. 557–560).
- Żbikowski, K., & Antosiuk, P. (2021). A machine learning, bias-free approach for predicting business success using crunchbase data. *Information Processing & Management*, 58(4), 102555.

- Zhang, C., Shi, W., Li, X., Zhang, H., & Liu, H. (2018). Improved bare pcb defect detection approach based on deep feature learning. *The Journal of Engineering*, 2018(16), 1415–1420.
- Zhou, Y., Yuan, M., Zhang, J., Ding, G., & Qin, S. (2023). Review of vision-based defect detection research and its perspectives for printed circuit board. *Journal of Manufacturing Systems*, 70, 557–578.
- Zingade, D. S., Deshmukh, R. K., & Kadam, D. B. (2023). Sentiment analysis using multi-objective optimization-based feature selection approach. In *2023 4th international conference for emerging technology (incet)* (pp. 1–6).