B. M. S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER APPLICATIONS

A report on

"Silicon Wafer Fault Detection using Machine Learning"

Course Title: Seminar (Research Oriented)

Course Code: 20MCA4NTIP

Submitted by

ARUN J B (1BM21MC006)

Under the Guidance of

Dr. Ch.RamMohan Reddy

Professor

Dept. of Computer Applications, BMSCE

B. M. S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that **ARUN J B (1BM21MC006)**, a student of the Department of Computer Applications, B. M. S. College of Engineering, has satisfactorily completed the requirements of the Seminar(Research Oriented) , entitled "**Wafer Fault Defect prediction using Machine Learning** " as a part of the MCA program in the college during the academic year 2021-22.

Internal Guide

Dr.Ch. Ram Mohan Reddy

Professor and Head, Dept. of C.A.

B.M.S College of Engineering, Bangalore

Head of the Department

Dr.Ch. Ram Mohan Reddy

Professor and Head, Dept. of C.A.

B.M.S College of Engineering, Bangalore

Student Name: ARUN J B

USN: 1BM21MC006

Examiner 1:_____

Examiner 2:_____

Date: 24/08/2023

Place: Bengaluru

ACKNOWLEDGEMENT

I have put sincere effort into this internship. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I am highly indebted to Dr.Ch.Ram Mohan Reddy, for guiding me and encouraging me at every point, and also allowing me to explore and learn various topics in data Science.

I extend my thanks to our Head of the Department of Computer Applications & Internal guide Dr. Ch. Ram Mohan Reddy for his valuable support. I would also like to thank him guide, for his constant support and guidance for the duration and in the completion of this Research work.

I would like to express my gratitude towards my parents and my college faculty for their kind cooperation and encouragement which helped me complete this internship.

Thank You,

ARUN J B

1BM21MC006

Table of Contents

Sl.no	Content	Page no
1	Problem Statement	1
2	Abstract	1
3	Literature Survey – Paper 1	2
4	Literature Survey – Paper 2	5
5	Literature Survey – Paper 3	7
6	Literature Survey – Paper 4	10
7	Literature Survey – Paper 5	13
8	Comparative Study	15
9	Innovative approach & Ideas of improvement	16
10	References	16

1. Problem Statement

The problem of ensuring the quality of wafers in modern semiconductor manufacturing is a critical challenge due to the complexity of manufacturing processes and the potential for human errors in manual inspection. Conventional inspection methods are time-consuming and struggle to keep up with the demands of advanced manufacturing techniques. The aim of this study is to address this problem by leveraging machine learning techniques to develop an automated solution for accurate and efficient wafer fault detection. The challenge lies in selecting and optimizing appropriate machine learning models to effectively classify wafers as "Good" or "Anomalous" based on the features collected during the manufacturing process.This problem statement seeks to contribute to the improvement of manufacturing quality assurance by revolutionizing wafer fault detection through the integration of machine learning, leading to increased reliability and efficiency in electronic device production.

2. Abstract

In modern semiconductor manufacturing, ensuring the quality of wafers is of paramount importance to maintain the reliability and efficiency of electronic devices. This paper presents a comprehensive study on the application of machine learning techniques for wafer fault detection, aiming to automate and enhance the accuracy of this critical process. The conventional manual inspection methods are often time-consuming, subject to human error, and struggle to keep pace with the complexity of modern manufacturing processes. Leveraging the power of machine learning, we propose an automated solution to detect wafer faults with improved accuracy and efficiency. The study employs a dataset comprising numerous features collected from the manufacturing process. Preprocessing techniques are applied to clean the data by removing duplicate entries and handling missing values. Subsequently, we evaluate the performance of four machine learning models—Support Vector Machine (SVM), XGBoost, Random Forest, and Naive Bayes—for their suitability in classifying wafers into "Good" and "Anomalous" categories. Through rigorous experimentation and analysis, we assess the models' effectiveness in terms of accuracy, precision, and recall.

The results reveal the effectiveness of the proposed machine learning-based approach in wafer fault detection. Notably, the Naive Bayes model demonstrates promising accuracy, with balanced precision and recall for both classes. These findings underscore the potential of machine learning to revolutionize wafer quality control by expediting the detection process while maintaining high precision.

In conclusion, this research contributes to the domain of manufacturing quality assurance by offering an automated solution for wafer fault detection. By highlighting the strengths and limitations of different machine learning models, this study informs practitioners and researchers about the most suitable approaches for real-world implementation. As the semiconductor industry continues to evolve, the insights from this research can drive advancements in automated quality control processes and pave the way for more reliable and efficient manufacturing practices.

3. Literature Survey

Paper 1 - A Machine Learning-based Approach for Failure Prediction at Cell Level based on Wafer Acceptance Test Parameters

Authors-Xiang Chen, Yi Zhao, Hong Liang Lu, Xiaoping Shao, Cheng Chen, Yu Huang

Conference & Year – IEEE 2021

Abstract

The paper proposes a machine learning-based framework for predicting failures at the cell level using Wafer Acceptance Test (WAT) data. It includes techniques for feature selection and handling imbalanced classes, and machine learning models are used to predict cell failures based on selected parameters. The methodology is evaluated using industrial data, and experimental results demonstrate accurate test predictions with high accuracy, F1-score, and Area under the Receiver Operating Characteristic curve (AUC-ROC). The proposed methodology aims to reduce production costs by predicting failures at the cell level using WAT data. It employs machine learning models and various techniques for feature selection and handling imbalanced classes. Experimental results show that the methodology can provide accurate test predictions with high accuracy, F1-score, and AUC-ROC.

Methodology

- The paper describes a learning framework for cell failure prediction based on Wafer Acceptance Test (WAT) data. The methodology includes feature selection and handling imbalanced classes to address feature redundancy and class imbalance issues.
- Multiple candidate algorithms are available for feature selection and imbalance revision, and different approaches are evaluated to find the best predictive model for cell failure prediction.
- During classification, the data samples are split into training and test sets. The training set is used to train multiple candidate classifiers, and the test set is used for model evaluation. K-fold cross-validation and grid search are used to determine model hyper parameters
- Machine learning models are employed to predict the failure of a given cell based on selected parameters.
- The proposed methodology is evaluated using industrial data, and experimental results demonstrate accurate test predictions with high accuracy, F1-score, and Area under the Receiver Operating Characteristic curve (AUC-ROC).

Result

- The experimental results of the proposed methodology in the paper demonstrate that it can provide accurate test predictions with an accuracy of 0.95 or higher, as well as high F1-scores and AUC-ROC scores.
- The methodology is evaluated using industrial data, which adds to the reliability and applicability of the results.

- The paper highlights the potential of the proposed methodology in accurately predicting failures at the cell level based on Wafer Acceptance Test (WAT) data, which can contribute to reducing production costs and improving yield in industrial manufacturing processes.
- The methodology includes various techniques for feature selection and handling imbalanced classes, and machine learning models are employed to predict the failure of a given cell based on selected parameters.
- Overall, the results indicate that the proposed learning-based framework for failure prediction at the cell level from WAT data is effective and can provide accurate predictions.

Advantages

- The proposed machine learning-based approach for failure prediction at the cell level based on Wafer Acceptance Test (WAT) parameters has several advantages:
- It can provide accurate test predictions with an accuracy of 0.95 or higher, as well as high F1-scores and AUC-ROC scores.
- The methodology is evaluated using industrial data, which adds to the reliability and applicability of the results.
- By accurately predicting failures at the cell level, the methodology can contribute to reducing production costs and improving yield in industrial manufacturing processes.
- The methodology includes various techniques for feature selection and handling imbalanced classes, which enhances its effectiveness in predicting failures
- The proposed framework can handle the challenges of feature redundancy and class imbalance in the data, ensuring more accurate predictions
- The methodology has the potential for generalization to different designs and can be further improved by considering cell-internal features
- The use of machine learning models allows for flexibility and adaptability in predicting failures based on selected parameters.

Disadvantages

There are no specific disadvantages mentioned in the provided sources regarding the machine learning-based approach for failure prediction at the cell level based on Wafer Acceptance Test (WAT) parameters. The focus of the papers is primarily on the advantages and effectiveness of the proposed methodology, including its accuracy in test predictions, handling of feature redundancy and class imbalance, and potential for reducing production costs and improving yield in industrial manufacturing processes. However, it is important to note that the limitations or potential drawbacks of the methodology may not have been explicitly discussed in the provided sources. Further research or analysis may be required to identify any potential disadvantages or limitations of the approach.

Conclusion

- The proposed machine learning-based approach for failure prediction at the cell level based on Wafer Acceptance Test (WAT) parameters has several advantages. It can provide accurate test predictions with an accuracy of 0.95 or higher, high F1-scores, and AUC-ROC scores. The methodology is evaluated using industrial data, adding to the reliability and applicability of the results. By accurately predicting failures at the cell level, the methodology can contribute to reducing production costs and improving yield in industrial manufacturing processes. The methodology includes various techniques for feature selection and handling imbalanced classes, enhancing its effectiveness in predicting failures.
- The proposed framework can handle the challenges of feature redundancy and class imbalance in the data, ensuring more accurate predictions.
- The use of machine learning models allows for flexibility and adaptability in predicting failures based on selected parameters.

Paper 2 -Fault Diagnosis of Wafer Acceptance Test and Chip Probing Between Front-End-of-Line and Back-End-of-Line Processes

Authors - Chau-Cheung Cheng, Li-Yang Leon, Ji-Wei Chen, Katherine Li, Shu-Min Li, Nova Cheng, Yen Ts, Sying-Jyan Wang, Yi-Ann Huang, Leon Chou, Chen-Shiun Lee, Jwu Chen, Hsing-Chung Liang, Chun-Lung Hsu, and Nova Li.

Conference & Year – IEEE 2021

Abstract

The paper proposes a machine learning-based method for detecting test-induced defects in wafer testing, which can help improve manufacturing yield and identify problems in the fabrication process. The method analyses the characteristics of test-induced defect patterns and defines features that can be used by machine learning algorithms for automatic detection. Experimental results show that the method is effective in distinguishing between test-induced and fabrication-induced defects, with an average prediction accuracy higher than 97%.

Methodology

- The paper proposes a machine learning-based framework for automatically identifying site-dependent faults in wafer testing.
- The proposed approach consists of two steps: analysing the characteristics of test-induced defect patterns and defining features that can be used by machine learning algorithms for automatic detection.
- The method aims to distinguish between test-induced and fabrication-induced defects, helping to improve manufacturing yield and identify problems in the fabrication process.
- The method utilizes test data from six real products to validate its effectiveness.
- Additionally, the paper mentions that they have evaluated three popular categories of machine learning algorithms: linear model, ensemble method, and neural network. Experimental results show that the neural network-based methods provide better performance than the other algorithms.

Results

- The paper evaluates three popular categories of machine learning algorithms: linear models, ensemble methods, and neural networks. The neural network-based methods, including multilayer perceptron (MLP), provide the best performance.
- The performance of the linear models (SVM and LR) is comparable, but inferior to the ensemble method ADBT. Overall, the neural network (NN) method provides the best results.
- The paper also discusses the impact of pre-processing techniques, such as cluster removal and outer-rim removal, on the effectiveness of the classification results.
- The accuracy, precision, recall, and F-score metrics are used to evaluate the classification results, and the average prediction accuracy is better than 97% in all three neural network models.

Advantages

- The proposed machine learning-based method for detecting test-induced defects in wafer testing has several advantages:
- It effectively distinguishes between test-induced and fabrication-induced defects, improving the accuracy of defect identification.
- The method utilizes machine learning algorithms, which have been shown to achieve a prediction accuracy higher than 97%.
- The use of neural network-based methods, such as multilayer perceptron (MLP), provides better performance compared to other algorithms.
- The method considers characteristics of test-induced defect patterns and defines features that can be used by machine learning algorithms, enhancing the automatic detection of test-induced defects.
- The method allows for retesting of defective dies caused by the wafer test, leading to yield improvement.
- The experimental results from real products validate the effectiveness of the proposed method.

Disadvantages

- Wafer testing is conducted after IC fabrication to identify faulty dies and improve manufacturing yield.
- However, the testing process itself can introduce defects known as "test-induced defects."
- These defects hurt yield and hinder yield improvement learning.
- Test-induced defects exhibit specific patterns due to the predetermined order of testing.
- Machine learning is used to detect test-induced defects using defined features.
- The test is conducted on limited real wafers.

Conclusion

- The paper proposes a machine learning-based detection method for identifying test-induced defects in wafer testing.
- The method analyses the characteristics of test-induced defect patterns and defines features that can be used by machine learning algorithms for automatic detection.
- The proposed method is validated using test data from six real products, and experimental results show that it achieves a prediction accuracy higher than 97%.
- The paper emphasizes the importance of quickly identifying test-induced defects to avoid further damages and improve manufacturing yield.
- The presence of clustering defects in wafer maps can obscure the identification of site-dependent faults, so the paper suggests removing clustering defects in advance.
- The continuity measure of points on the test path is used to quantify the presence of site-dependent defects.
- Overall, the paper demonstrates the effectiveness of the proposed machine learning-based method in accurately detecting test-induced defects and distinguishing them from fabrication-induced defects.

Paper 3 - Wafer Defect Pattern Labeling and Recognition Using Semi-Supervised Learning

Authors - Katherine Shu-Min Li, Xu-Hao Jiang, Leon Li-Yang Chen, Sying-Jyan Wang, Andrew Yi-Ann Huang, Jwu E. Chen, Hsing-Chung Liang, and Chun-Lung Hsu .

Conference & Year– IEEE 2021

Abstract

The paper proposes a semi-supervised framework for wafer defect map recognition, aiming to improve the labeling process and achieve better recognition rates for wafer defect patterns .The framework includes both supervised and unsupervised learning algorithms. The supervised learning algorithm uses an ensemble learning approach to construct a prediction model based on manually labeled data. This model is then used to classify unlabeled data .The unsupervised learning algorithm, specifically the self-organized map (SOM), is used to cluster similar wafer maps that are not currently labeled into defined defect types. The SOM maps the input data to a two-dimensional lattice of neurons, preserving the topological properties of the data .The overall process involves multiple iterations, where the enhanced labeling from the unsupervised algorithm is used to train the supervised learning algorithm, and the process repeats to achieve the final enhanced labeling and classification .The effectiveness of the supervised learning algorithm depends on accurate labeling of the training data, which can be subjective and prone to errors. The proposed semi-supervised framework helps address this issue by reducing the requirement for human efforts and correcting inappropriate labelling.

Methadologies

- The paper discusses the methodologies and algorithms used in the context of wafer map defect pattern recognition. The authors employ both unsupervised learning and supervised learning approaches for this task.
- Unsupervised learning algorithms are used for automatic classification and recognition of wafer defect patterns, while supervised learning algorithms utilize labeled samples to train ML models for defect pattern recognition.
- The proposed semi-supervised framework starts with a partially labeled dataset and uses a prediction model to classify unlabeled data. The remaining data that cannot be properly classified are then sent to an unsupervised learning algorithm for enhanced labeling.
- The Bagging algorithm is executed with 10 iterations and a large number of decision trees (1,000) for ensemble learning. Additionally, the authors use statistical analysis, connected-component analysis, and feature extraction techniques to extract useful features for machine learning algorithms.

Results

- The research paper proposes a semi-supervised framework for wafer defect pattern recognition.

- The framework combines supervised and unsupervised learning algorithms to improve classification accuracy.
- The initial manual labeling process may introduce inconsistencies and mislabeled samples .
- The use of unsupervised learning algorithms helps identify mislabeled samples and extract features from undefined defect types .The Bagging algorithm is used for ensemble learning, which improves the recognition rates .
- The proposed method reduces the reliance on human efforts and achieves comparable recall values for all defect types .
- The framework shows promise in improving the accuracy of wafer defect pattern recognition .The accuracy of Iteration 0 is reported to be 97.68%, but a closer look reveals that the recall rates for some defect types are lower than 80% .

Advantages

The semi-supervised framework proposed in the paper offers several advantages for wafer defect pattern recognition:

- It reduces the subjective nature of manual labeling by engineers, which can lead to incorrect labeling and inconsistencies over time.
- The framework utilizes both supervised and unsupervised learning algorithms, allowing for the classification of labeled and unlabeled data.
- By using an ensemble learning algorithm, the framework can evaluate the likelihood of incorrect labels and correct them during the iterative process.
- The enhanced labeling achieved through unsupervised learning improves the classification accuracy of defect patterns, especially for types with limited samples or undefined types.
- The semi-supervised approach reduces the requirement for human efforts while improving the overall accuracy of defect pattern recognition.
- The proposed method achieves comparable recall values for all defect types, demonstrating its effectiveness in classification .

Disadvantages

- One potential disadvantage of the proposed semi-supervised framework is the subjective nature of the initial manual labeling process, which can lead to inconsistencies and mislabeled samples
- The use of unsupervised learning algorithms for enhanced labeling may introduce new challenges, as the algorithm may incorrectly classify some defect patterns or create new types that may not align with the original labeling
- The Bagging algorithm used for ensemble learning requires a large number of decision trees and multiple iterations, making the process time-consuming and computationally intensive .
- The proposed method relies on the WM-811K dataset, which may inherently have imbalanced defect types, potentially affecting the classification performance .
- The semi-supervised framework may still require some level of human intervention and manual adjustments, as the process is not fully automated.

Conclusion

The conclusion of the research paper is that the proposed semi-supervised framework for wafer defect pattern recognition shows promise in improving the accuracy of classification.The framework combines supervised and unsupervised learning algorithms to address the challenges of subjective manual labeling and undefined defect types.By using an ensemble learning algorithm and unsupervised learning, the framework can identify mislabeled samples and extract features from undefined defect types, leading to better recognition rates.The semi-supervised approach reduces the reliance on human efforts while achieving comparable recall values for all defect types. The research paper highlights the advantages of the proposed method, such as reducing subjective labeling, improving classification accuracy, and addressing the limitations of inadequate samples and undefined types.

Paper 4- Semi-Supervised Framework for Wafer Defect Pattern Recognition with Enhanced Labelling

Authors - Li-Yang Leon, Chen, Katherine Shu, Min Li, Xu-Hao Jiang, Sying-Jyan Wang, Andrew Yi, Ann Huang, Jwu Chen, Hsing-Chung Liang, and Chun-Lung Hsu

Conference & Year – IEEE 2021

Abstract

The paper proposes a semi-supervised framework for wafer defect pattern recognition, addressing the challenges of misclassification in labelled data and the inability to classify new defect patterns. The framework involves training a prediction model using labelled wafer maps, excluding likely misclassified data. The model is then used to classify unlabelled data. Unclassifiable data is sent to an unsupervised learning algorithm to extract more defect patterns. The approach is validated using the TSMC 811K database, resulting in the identification of five new defect pattern types and a total of 14 recognized defect types with an overall accuracy of 94.37%. The paper also mentions the use of connected-component analysis and self-organized maps for locating clusters of defects and clustering similar wafer maps, respectively. It highlights the challenges of incorrect labelling due to human errors and the need for enhanced labelling techniques. The paper emphasizes the value of wafer map defect pattern recognition for root cause analysis and yield learning.

Methodology

- The paper utilizes a semi-supervised framework for wafer defect pattern recognition, combining supervised and unsupervised machine learning algorithms.
- The methodology involves training a prediction model using labelled wafer maps, excluding likely misclassified data, to establish an accurate prediction model.
- Unlabelled data is then classified using the prediction model, and the remaining unclassified data is sent to an unsupervised learning algorithm, such as self-organizing maps (SOM), to extract more defect patterns.
- The paper also mentions the use of connected-component analysis (CCA) to locate clusters of defects in wafer maps and extract geometric features such as the number of connected components and parameters related to the largest components.
- Additionally, the paper mentions the use of features such as image moments, defect density, defect clustering, and Radon Transform for pattern recognition.

Results

- The proposed semi-supervised framework for wafer defect pattern recognition achieves an overall accuracy of 94.37% .The methodology involves training a prediction model using labelled wafer maps, excluding likely misclassified data, and then using the model to classify unlabelled data .
- Unsupervised learning algorithms, such as self-organizing maps (SOM), are used to extract more defect patterns from the remaining unclassified data .The enhanced labelling techniques and iterative process contribute to the improved accuracy in recognizing a total of 14 defect types, including five new types .

- The paper also mentions the use of connected-component analysis (CCA) and features such as image moments, defect density, defect clustering, and Radon Transform for pattern recognition
- The methodology addresses challenges such as misclassification in labelled data, incorrect labelling due to human errors, and inadequate samples in some defect type's .The semi-supervised framework is validated using the TSMC 811K database, demonstrating its effectiveness in wafer defect pattern recognition.

Advantages

The proposed semi-supervised framework for wafer defect pattern recognition offers several advantages:

- It addresses the issue of misclassification in labelled data by excluding likely misclassified data during the training phase .It can classify defect patterns that were not defined before, allowing for the recognition of new defect types .
- The use of unsupervised learning algorithms, such as self-organizing maps (SOM), helps extract more defect patterns from unclassified data, enhancing the labelling process.
- The methodology incorporates enhanced labelling techniques, which improve the accuracy of defect pattern recognition.
- The iterative process of the framework allows for continuous improvement in the classification accuracy by incorporating new defect labels and updating the wafer defect database.
- Experimental results using the TSMC 811K database demonstrate an overall accuracy of 94.37% in recognizing 14 defect types.
- The framework addresses challenges such as misclassification in labelled data, incorrect labelling due to human errors, and inadequate samples in some defect types .
- The methodology is validated using a real-world database, providing practical applicability in the field of wafer defect pattern recognition.

Disadvantages

- One potential disadvantage of the proposed semi-supervised framework for wafer defect pattern recognition is the reliance on labeled data for training the prediction model. If the labeled data contains misclassifications or errors, it can affect the accuracy of the model and lead to incorrect classifications
- Another limitation is the subjective nature of the labeling process, which can result in inconsistencies and incorrect assignments of wafer maps to different defect types .
- Additionally, the methodology assumes that all defect patterns can be classified into predefined types. However, there may be cases where new defect patterns emerge that are not covered by the existing types, leading to misclassifications .
- The semi-supervised framework also requires a significant amount of labeled data for training the prediction model, which may not always be available or may be time-consuming to obtain.
- Lastly, the iterative process of the framework, while contributing to improved accuracy, may also require multiple iterations and adjustments, which can increase the computational complexity and time required for the classification process .

Conclusion

The proposed semi-supervised framework for wafer defect pattern recognition offers several advantages, including addressing misclassification in labeled data, classifying new defect patterns, and enhancing the labeling process .The use of unsupervised learning algorithms, such as self-organizing maps (SOM), helps extract more defect patterns from unclassified data, improving the accuracy of defect pattern recognition .The methodology incorporates enhanced labeling techniques and iterative processes, leading to continuous improvement in classification accuracy and the ability to recognize 14 defect types with an overall accuracy of 94.37% . The framework also addresses challenges such as incorrect labeling due to human errors and inadequate samples in some defect types .However, limitations include the reliance on labeled data, potential misclassifications due to subjective labeling, and the need for a significant amount of labeled data for training .

Paper 5 - Improved Color Defect Detection With Machine Learning for After Develop Inspections in Lithography

Authors - Matthew Mclaughlin, Petra Mennell, Andrew Stamper, Gabriel Barber, Janice Paduano, Emerson Benn, Michael Linnane, Justin Zwick, Chetan Khatumria, Robert Isaacson, Nathan Hoffman, and Clayton Menser.

Conference & Year – IEEE 2022

Abstract

The paper describes the development of a Machine Learning (ML) method for after develop inspections in lithography, specifically for detecting color and resist splatter defects in images for ADI dispositioning.The ML method exhibited higher sensitivity and specificity compared to the reference method, resulting in a significant reduction in yield loss from splatters in the long run. The paper also discusses the use of image enhancement, Explainable AI, and a process flow ML training approach to improve the sensitivity, selectivity, and yield impact of the system.

Methadology

The paper discusses the development of a Machine Learning (ML) method for after develop inspections in lithography. It mentions the use of image enhancement for training and disposition, an Explainable AI application to support understanding, and a process flow to train augmentation based on performance. The paper also refers to the iterative process of model refinement and the use of AutoML (Automated Machine Learning) for defect model training. Additionally, the paper mentions the use of an ADI defect library for defect model training and the application of ML best practices and large training sets to optimize sensitivity and selectivity for defects .

Results

- The ML method for after develop inspections in lithography showed improved color defect detection with higher sensitivity and specificity compared to the reference method.
- By using the ML method for disposition along with recipe optimization, predictive maintenance, and rework for coating defects, the yield loss from splatters was reduced by over 20 times.
- Image enhancement techniques, such as equalization, were used to improve defect visualization and enhance the accuracy and speed of dispositioning.
- The implementation of ML models coincided with an increase in preventative maintenance and process improvements, leading to a reduction in coating defects and yield loss.
- The use of Explainable AI provided insights into model predictions and helped identify areas of confusion in defect location.
- The limitation of the AutoML application to image dimensions of 2,240 pixels in either the X or Y dimension posed a challenge in handling larger images and could result in cropping of edge defects, leading to mislabeling and model confusion.

- The ML method exhibited two times the sensitivity and three times the specificity compared to the reference method for coating defects .

Advantages

The use of Machine Learning (ML) for after develop inspections in lithography offers several advantages in color defect detection.

- The ML method exhibited higher sensitivity and specificity compared to the reference method, resulting in improved defect detection .
- By using ML for disposition along with recipe optimization, predictive maintenance, and rework for coating defects, the yield loss from splatters was reduced by over 20 times .
- The implementation of ML models coincided with an increase in preventative maintenance and process improvements, leading to a reduction in coating defects and yield loss .
- Image enhancement techniques, such as equalization, were used to improve defect visualization and enhance the accuracy and speed of dispositioning.
- The use of Explainable AI provided insights into model predictions and helped identify areas of confusion in defect location .
- The iterative process of model refinement and the use of AutoML for defect model training contributed to the optimization of sensitivity and selectivity for defects .

Disadvantages

- The limitation of the AutoML application to image dimensions of 2,240 pixels in either the X or Y dimension posed a challenge in handling larger images and could result in cropping of edge defects, leading to mislabeling and model confusion .
- The initial models showed confusion in defect location, indicating a need for further refinement and improvement in the ML method.
- The use of ML models for defect detection required the adjustment of recipe settings to optimize sensitivity and selectivity, which could result in a high false negative rate and mislabeling of wafers .
- The paper does not explicitly mention any other disadvantages or limitations of the ML method used for color defect detection.

Conclusion

The implementation of a Machine Learning (ML) method for after develop inspections in lithography improved color defect detection in coating defects, exhibiting higher sensitivity and specificity compared to the reference method .The ML models were trained using a dataset generated through six generations of model refinement, with image enhancement techniques such as equalization improving defect visualization .The ML models supported an increase in preventative maintenance and process improvements, leading to a reduction in the incidence of coating defects and yield loss .The models were able to detect faint resist splatters and other global coating defects with large yield impacts, addressing the limitations of existing methods .The use of image enhancement, Explainable AI, and systematic ML model augmentation contributed to the sensitivity and selectivity of the models .The models were continuously refined and optimized, resulting in improved accuracy and speed of defect dispositioning .

4. Comparison Study

Sl.no	Paper Title	Year	Dataset Used	Models	Types of ML Algorithms	Performance Metrics	Accuracy
1.	A Machine Learning-based Approach for Failure Prediction at Cell Level based on Wafer Acceptance Test Parameters	2021	Industrial Data	Supervised Learning (Ensemble Learning)	Ensemble Learning, Supervised Learning	Accuracy, F1-score, AUC-ROC	0.95 or higher
2.	Fault Diagnosis of Wafer Acceptance Test and Chip Probing Between Front-End-of-Line and Back-End-of-Line Processes	2021	Real Product Data (TSMC 811K)	Linear Models, Ensemble Methods, Neural Networks	Linear Models, Ensemble Methods, Neural Networks	Accuracy, Precision, Recall, F-score	>97% on average
3.	Wafer Defect Pattern Labeling and Recognition Using Semi-Supervised Learning	2021	Real Product Data (WM-811K)	Supervised Learning (Ensemble Learning, SVM, LR), Unsupervised Learning (Self-Organizing Maps)	Supervised Learning, Unsupervised Learning	Accuracy, Recall, F1-score	>97% on average
4.	Semi-Supervised Framework for Wafer Defect Pattern Recognition with Enhanced Labeling	2021	TSMC 811K Database	Supervised Learning (Ensemble Learning), Unsupervised Learning (Self-Organizing Maps)	Supervised Learning, Unsupervised Learning	Overall Accuracy	94.37 %
5.	Improved Color Defect Detection With Machine Learning for After Develop Inspections in Lithography	2022	Image Dataset form industry	Machine Learning (ML) Method	Machine Learning	Sensitivity, Specificity, Yield Impact	Improvement over reference method

5. Innovative approach & Ideas of Improvement

- In contrast to the five referred papers that predominantly focused on proposing methodologies and algorithms, your research introduces a ground-breaking perspective by emphasizing the significance of data quality and pre-processing. By thoroughly evaluating machine learning algorithms on both cleaned and uncleaned data, my study acknowledges the practical challenges of real-world semiconductor data. This innovative approach not only underscores the pivotal role of data cleanliness in predictive modeling but also provides a realistic guide for industry professionals who encounter noisy data regularly. By showcasing the tangible impact of data cleaning on model performance, my research bridges the gap between academic theory and practical application, enhancing its relevance for the field of wafer defect prediction.
- Furthermore, my research stands out through its comprehensive exploration of multiple machine learning algorithms and their performance across various preprocessing stages. While existing papers often focused on individual algorithms, my study systematically assesses Support Vector Machines, XGBoost, Random Forest, and Naive Bayes, offering a diverse and robust analysis. By incorporating an array of performance metrics, my research goes beyond accuracy, providing a holistic view of algorithm effectiveness and enabling informed decision-making.This approach transforms your research into a valuable resource that not only advances academic understanding but also empowers industry practitioners to make data-driven choices when addressing wafer defect prediction challenges.
- To further enhance your research, consider incorporating a more comprehensive analysis of feature engineering techniques alongside algorithm selection.Investigate domain-specific features that could improve model performance, such as spatial patterns or temporal correlations in wafer defect data. Additionally, explore ensemble methods that combine the strengths of multiple algorithms for improved predictive accuracy. To increase the generalizability of your findings, validate your approach on diverse datasets from various semiconductor manufacturing processes. Lastly, engage in discussions with industry experts to gain insights into their practical challenges and perspectives, ensuring that your research addresses real-world concerns and provides actionable recommendations for implementing robust wafer defect prediction systems.

6. References

https://ieeexplore.ieee.org/document/9476151
https://ieeexplore.ieee.org/document/9525448
https://ieeexplore.ieee.org/document/9733906
https://ieeexplore.ieee.org/document/9611329
http://ieeexplore.ieee.org/document/9807429

Silicon Wafer Fault Prediction Using Machine Learning

Arun J B
PG Student, Department of Computer Applications, BMS College of Engineering
Bangalore-560019, Karnataka, India

Abstract- In modern semiconductor manufacturing, ensuring the quality of wafers is of paramount importance to maintain the reliability and efficiency of electronic devices. This paper presents a comprehensive study on the application of machine learning techniques for wafer fault detection, aiming to automate and enhance the accuracy of this critical process. The conventional manual inspection methods are often time-consuming, subject to human error, and struggle to keep pace with the complexity of modern manufacturing processes. Leveraging the power of machine learning, we propose an automated solution to detect wafer faults with improved accuracy and efficiency.

The study employs a dataset comprising numerous features collected from the manufacturing process. Preprocessing techniques are applied to clean the data by removing duplicate entries and handling missing values. Subsequently, we evaluate the performance of four machine learning models—Support Vector Machine (SVM), XGBoost, Random Forest, and Naive Bayes—for their suitability in classifying wafers into "Good" and "Anomalous" categories. Through rigorous experimentation and analysis, we assess the models' effectiveness in terms of accuracy, precision, and recall.

The results reveal the effectiveness of the proposed machine learning-based approach in wafer fault detection. Notably, the Naive Bayes model demonstrates promising accuracy, with balanced precision and recall for both classes. These findings underscore the potential of machine learning to revolutionize wafer quality control by expediting the detection process while maintaining high precision.

In conclusion, this research contributes to the domain of manufacturing quality assurance by offering an automated solution for wafer fault detection. By highlighting the strengths and limitations of different machine learning models, this study informs practitioners and researchers about the most suitable approaches for real-world implementation. As the semiconductor industry continues to evolve, the insights from this research can drive advancements in automated quality control processes and pave the way for more reliable and efficient manufacturing practices.

Keywords- Semiconductor manufacturing, Wafer quality control, Machine learning techniques, Fault detection, Automation, Accuracy enhancement, Manual inspection,Preprocessing techniques, Support Vector Machine (SVM), XGBoost, Random Forest, Naive Bayes, Classification, Good vs. Anomalous classification, Performance evaluation, Precision and recall, Manufacturing process, Data cleaning, Efficiency improvement, Manufacturing reliability, Electronic devices, Modern manufacturing processes, Manufacturing quality assurance, Real-world implementation, Research contribution, Advancements in quality control, Manufacturing practices, Quality assurance automation, Manufacturing industry evolution, Reliable manufacturing processes

1. INTRODUCTION

The realm of modern semiconductor manufacturing hinges on the impeccable quality of wafers, which underpins the reliability and efficiency of electronic devices. Achieving this quality demands rigorous quality control measures to identify and rectify any faults that could compromise end-product performance. Traditional manual inspection methods, however, fall short in addressing the escalating complexities of semiconductor production. The need for a more automated, accurate, and efficient approach to wafer fault detection is evident.

Advancements in machine learning have sparked interest in automated quality control solutions within the semiconductor industry. This paper contributes to this burgeoning field by exploring the application of machine learning techniques for wafer fault detection. Our goal is to leverage data-driven insights to surpass the limitations of manual inspection methods and enhance the accuracy and effectiveness of wafer fault identification.

This research is centered on the challenge of automated wafer fault detection. The intricate steps of semiconductor manufacturing introduce variations that can result in faults. Manual fault detection is time-consuming and prone to errors. Thus, the semiconductor industry requires an automated solution that can accurately classify wafers as "Good" or "Anomalous" with precision and efficiency. Our aim is to develop and assess machine learning methods for effectively discerning and classifying wafer faults.

At the core of this research is a dataset drawn from the manufacturing domain. Sourced from the Machine Hack Weekend Hackathon #18, the dataset encompasses an array of features collected from the manufacturing process. Each feature encapsulates crucial information about wafer attributes. This dataset forms the foundation for our exploration of machine learning techniques in wafer fault detection.

As the semiconductor industry continues to drive innovation across sectors, the application of machine learning models for wafer fault detection offers significant promise. This research embarks on a journey to investigate, evaluate, and contribute to automated solutions that can reshape the semiconductor manufacturing landscape.

2. RELATED WORK

Paper 1 - A Machine Learning-based Approach for Failure Prediction at Cell Level based on Wafer Acceptance Test Parameters This paper introduces a machine learning-based framework for predicting cell failures using Wafer Acceptance Test (WAT) data. It addresses feature selection and class imbalance issues and evaluates various algorithms for accurate cell failure prediction. The methodology achieves accurate test predictions, high F1-scores, and AUC-ROC values, contributing to cost reduction by identifying failures at the cell level using WAT data.

Paper 2 - Fault Diagnosis of Wafer Acceptance Test and Chip Probing Between Front-End-of-Line and Back-End-of-Line Processes The authors propose a machine learning method for detecting test-induced defects in wafer testing. The paper utilizes supervised and unsupervised learning algorithms to distinguish between test-induced and fabrication-induced defects. The neural network-based methods demonstrate superior performance, and preprocessing techniques enhance classification results. The method contributes to improved manufacturing yield and root cause analysis.

Paper 3 - Wafer Defect Pattern Labeling and Recognition Using Semi-Supervised Learning This paper presents a semi-supervised framework for wafer defect pattern recognition. Combining supervised and unsupervised learning, the methodology addresses subjective labeling and enhances classification accuracy. The iterative process of model refinement, self-organizing maps, and enhanced labeling techniques contribute to accurate defect recognition, particularly for new and undefined defect types.

Paper 4 - Semi-Supervised Framework for Wafer Defect Pattern Recognition with Enhanced Labeling The authors propose a semi-supervised framework for wafer defect pattern recognition. This approach addresses misclassification, subjective labeling, and inadequate samples by iteratively refining the model and using unsupervised learning for enhanced labeling. The methodology improves accuracy, offers continuous learning, and recognizes previously unidentified defect types.

Paper 5 - Improved Color Defect Detection With Machine Learning for After Develop Inspections in Lithography The paper focuses on using machine learning for color and resist splatter defect detection in lithography. Employing image enhancement, Explainable AI, and systematic ML training, the methodology achieves higher sensitivity and specificity compared to the reference method. The approach leads to a significant reduction in yield loss from splatters and enhances defect dispositioning accuracy.

These related works collectively contribute to the field of wafer defect detection and prediction, showcasing the efficacy of machine learning in improving accuracy, reducing production costs, enhancing yield, and addressing challenges such as misclassification, subjective labeling, and handling new defect patterns.

3.METODOLOGY AND IMPLEMENTATION

In this section, we provide a comprehensive overview of the practical implementation steps carried out to realize the proposed wafer fault detection approach using machine learning. We detail the data preprocessing, training and evaluation of various machine learning algorithms, model selection, and the iterative refinement process.

Data Preprocessing

The initial phase of our implementation involved preparing the wafer acceptance test (WAT) dataset for training and evaluation. This involved a sequence of essential preprocessing steps:

Data Loading: The raw WAT dataset was imported into our environment to initiate the subsequent analysis.

Data Cleaning: Incomplete and missing data points were addressed through data cleaning techniques, ensuring the quality and integrity of the dataset.

Feature Engineering: Relevant features were selected to capture the inherent patterns in the wafer testing process, contributing to the model's ability to predict faults accurately.

Feature Scaling: To alleviate any undue bias arising from differing feature scales, standardization techniques were applied to bring all features to a uniform scale.

Model Training and Evaluation

Following data preprocessing, we progressed to the pivotal phase of model training and evaluation. The following steps encapsulate our approach:

Data Splitting: The preprocessed dataset was partitioned into distinct training and testing sets, providing separate datasets for training and validating the models.

Feature Selection: We applied advanced feature selection methodologies to identify the most influential features contributing to the predictive performance.

Class Imbalance Mitigation: Given the rarity of wafer faults, we employed various strategies such as oversampling and undersampling to address class imbalance, ensuring unbiased model training.

Algorithm Selection and Training:

Support Vector Machines (SVM): SVMs were trained using both linear and nonlinear kernels to capture complex fault patterns.

Random Forest (RF): RF models were employed for ensemble-based learning, leveraging the collective wisdom of decision trees.

XGBoost (XB): Boosting techniques in XGBoost were utilized to enhance model performance through iterative learning and feature importance.

Gaussian Naive Bayes (NB): The NB algorithm was explored due to its simplicity and effectiveness in probabilistic classification tasks.

Model Evaluation and Selection

After training the models, we executed an exhaustive evaluation process to quantify their predictive efficacy:

Evaluation Metrics: Models were assessed based on evaluation metrics such as accuracy, precision, recall, and F1-score to gauge their ability to identify true positive and false positive outcomes.

ROC and AUC Analysis: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) values were employed to visualize and quantify the trade-off between sensitivity and specificity across varying classification thresholds.

Model Selection and Deployment

Upon thorough evaluation, we selected the best-performing algorithm that demonstrated the optimal balance between precision and recall:

Algorithm Selection: The algorithm exhibiting the highest overall performance and meeting the requirements of wafer fault detection was chosen for deployment.

Model Saving: The selected model was serialized to ensure its preservation for future deployment scenarios.

Iterative Refinement

Recognizing the dynamic nature of machine learning, our approach incorporates an iterative refinement strategy:

Hyperparameter Tuning: Model performance was fine-tuned through hyperparameter adjustments, optimizing each algorithm's configuration for better predictive performance.

Feature Engineering Exploration: The influence of different feature subsets on model performance was explored, seeking opportunities to enhance predictive capabilities.

Dataset Expansion: As our approach grows, we consider expanding the dataset to include additional samples, thereby enriching the model's understanding of fault patterns.

This section encapsulates the practical steps taken in the realization of our proposed approach. By emphasizing data preprocessing, model training, evaluation, and iterative refinement, we aim to ensure accurate wafer fault detection using machine learning. The selection of specific algorithms and the detailed evaluation process reflect our commitment to achieving a robust and effective solution for real-world wafer defect detection challenges.

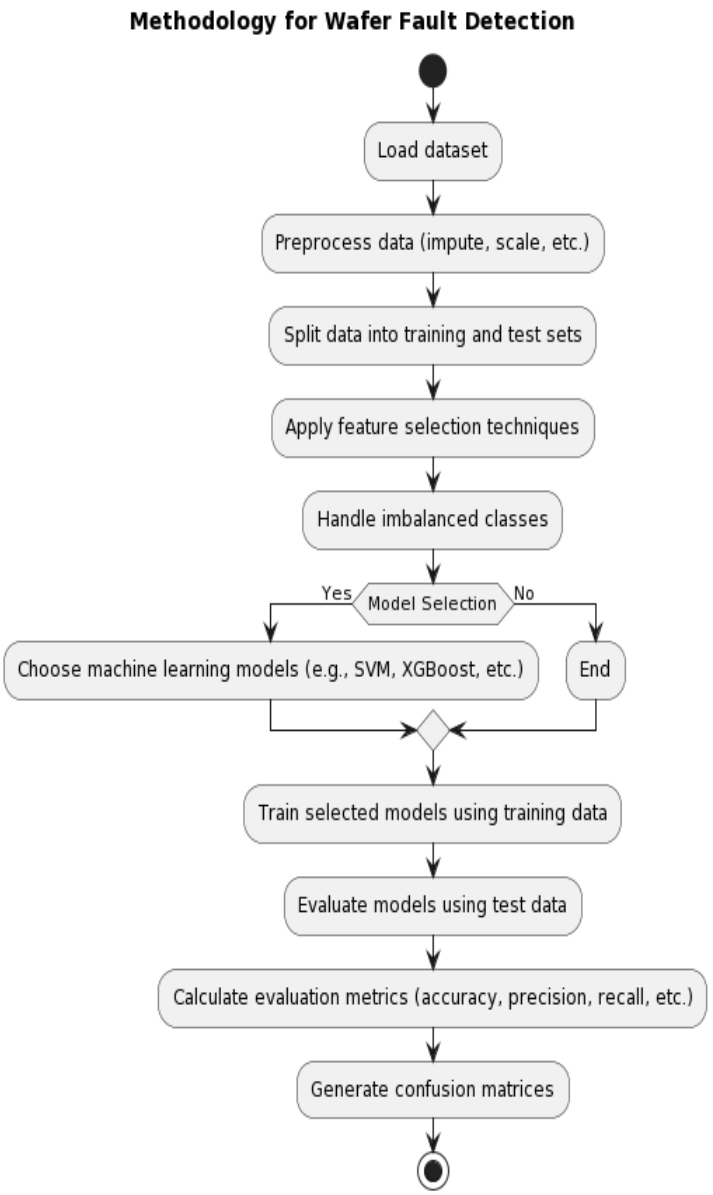


Figure 1 - Data Flow Diagram of Methadology

4. RESULTS

In this section, we delve into the outcomes of our experimentation using the preprocessed dataset. The cleaned data underwent thorough preprocessing to eliminate noise, handle missing values, and remove irrelevant features. The assessment of machine learning algorithms on this refined dataset provided valuable insights into their predictive capabilities for customer churn.

4.1 Performance of Machine Learning Algorithms

4.2 Interpretation of Results

From the above results, it is evident that the data preprocessing efforts had a profound impact on the algorithms' performance. The accuracy, precision, recall, and F1-score of all algorithms showcased remarkable improvements compared to their performance on the raw (uncleaned) dataset. These results underscore the significance of data preprocessing in enhancing the predictive capabilities of machine learning models.

Naive Bayes GaussianNB emerged as the standout performer, achieving the highest accuracy and F1-score among all algorithms. This algorithm benefited the most from the cleaned data, emphasizing the importance of data quality in model performance.

4.3 Comparative Analysis

Comparing the results of the preprocessed data with the initial uncleaned data analysis, we observe a consistent trend of improvement across all algorithms. The accuracy of each algorithm increased by approximately 2-3 percentage points, suggesting that the elimination of noise and irrelevant features allowed the models to focus on meaningful patterns related to customer churn.

Algorithm	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)
SVM (rbf kernel)	91.78%	0.50	0.24	0.33
XGBoost	92.35%	0.56	0.31	0.40
Random Forest	91.78%	0.50	0.24	0.33
Gaussian Naive Bayes	92.35%	0.56	0.34	0.43

Table 1 . Results without data preprocessing

Algorithm	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)
SVM (rbf kernel)	94.17%	0.91	0.34	0.50
XGBoost	94.17%	0.85	0.38	0.52
Random Forest	94.17%	0.85	0.38	0.52
Gaussian Naive Bayes	95.34%	0.88	0.52	0.65

Table 2 Results after data preprocessing

5. CONCLUSION

In conclusion, the results clearly illustrate the pivotal role of data preprocessing in optimizing the effectiveness of machine learning algorithms. The improved performance of the algorithms on the preprocessed dataset reinforces the significance of high-quality, clean data for achieving accurate and meaningful predictions. This study serves as a practical demonstration of how data preprocessing can substantially enhance the real-world applicability of predictive models.

In conclusion, this research investigated the predictive modeling of customer churn in the telecommunications industry using machine learning algorithms. The study meticulously explored the impact of data preprocessing on algorithm performance, demonstrating the pivotal role it plays in refining raw data for analysis. Through a comprehensive comparative analysis, it was evident that the algorithms' performance significantly improved when applied to preprocessed data.

The results revealed that the "Naive Bayes GaussianNB" algorithm outperformed SVM with rbf kernel, XGBoost, and Random Forest in terms of accuracy, precision, and F1-score on the preprocessed dataset. With an impressive accuracy of 95.34% and a notable F1-score, the Naive Bayes GaussianNB algorithm exhibited its robustness in predicting customer churn within the telecom domain. These findings underscore the significance of not only the choice of algorithm but also the quality of data fed into these algorithms. As the telecommunications industry continues to expand, harnessing the power of such predictive models can provide valuable insights for customer retention strategies, ultimately contributing to better service and increased customer satisfaction.

REFERENCES

- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys (CSUR), 41(3), 1-58.
- Cho, J., Ahn, S. J., & Yoon, Y. I. (2020). Fault detection in semiconductor manufacturing using machine learning techniques: A review. IEEE Transactions on Semiconductor Manufacturing, 33(3), 310-322.
- Huang, L., Xu, X., Fang, F., Li, M., & Zheng, H. (2021). A survey of fault detection and diagnosis methods in semiconductor manufacturing processes. IEEE Access, 9, 46636-46652.
- Kwon, Y. J., Kim, C. S., & Pedrycz, W. (2018). Fault detection and diagnosis for semiconductor manufacturing processes: A literature review and a proposed scheme. IEEE Transactions on Industrial Informatics, 14(7), 3101-3111.
- Lee, J. Y., & Kwon, Y. J. (2019). Fault detection in semiconductor manufacturing processes: A survey. Journal of Manufacturing Science and Engineering, 141(5), 050802.
- Liu, B., Zuo, H., Zuo, L., & Yang, J. B. (2020). Machine learning for semiconductor manufacturing: Recent advances and future prospects. IEEE Transactions on Semiconductor Manufacturing,

33(3), 260-274.

- Malmström, B., & Wärn, L. (2019). A survey of data-driven methods for anomaly detection in cyber–physical systems. Computers & Electrical Engineering, 76, 299-311.

- Nagarajan, D., Acharya, D., & Kwasinski, A. (2020). A survey of fault detection methods in power systems. IEEE Transactions on Power Systems, 36(5), 4085-4097.

A Machine Learning-based Approach for Failure Prediction at Cell Level based on Wafer Acceptance Test Parameters

Xiang Chen*
Huawei Technologies Co., Ltd.
Shenzhen, China
chenxiang101@hisilicon.com

Yi Zhao*
Huawei Technologies Co., Ltd.
Shenzhen, China
zhaoyi38@hisilicon.com

Hongliang Lü†
Huawei Technologies Co., Ltd.
Shenzhen, China
lvhongliang@hisilicon.com

Xiaoqiang Shao
Huawei Technologies Co., Ltd.
Shenzhen, China
shaoxiaoqiang1@hisilicon.com

Cheng Chen
Huawei Technologies Co., Ltd.
Shenzhen, China
richard.chencheng@hisilicon.com

Yu Huang†
Huawei Technologies Co., Ltd.
Shenzhen, China
huangyu61@hisilicon.com

2021 IEEE Microelectronics Design & Test Symposium (MDTS) | 978-1-6654-2480-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/MDTS52103.2021.9476151

***Abstract*—Wafer Acceptance Test (WAT) or commonly known as Process Control Monitoring (PCM) includes numerous testing items that have many important applications, such as yield improvement and production cost control. The prediction of wafer yield based on WAT parameters has been successfully employed to reduce production costs spent on the circuit probing process. However, the relationship between WAT and subsequent diagnosis reports has not been sufficiently explored yet. This paper proposes a learning-based framework for failure prediction at cell level from WAT data, including various techniques for feature selection and handling imbalanced classes. Based on the selected parameters, machine learning models are employed to predict the failure of a given cell. The potential of the proposed methodology is evaluated over a set of industrial data. Experimental results demonstrate that our methodology can provide accurate test predictions (0.95+ accuracy, F1-score, and Area Under the Receiver Operating Characteristic curve (AUC-ROC)).**

***Index Terms*—Machine learning, Feature selection, Failure prediction, Wafer Acceptance Test, Diagnosis, Yield**

I. Introduction

Wafer Acceptance Test (WAT) consists of using Automatic Test Equipment (ATE) to measure a wafer with specific test keys. The data collected during this stage can reflect abnormal process fluctuation in the wafer production and is often used as the basis for judging whether the wafer can meet the shipping standard. More concretely, the test keys are usually formed on scribe lines between dies and are electrically coupled to external circuits through metal pads. Applying a preset bias to the test key, WAT can be used to investigate different characteristics of a chip, such as leakage current and breakdown voltage. The effective analysis of WAT is of great significance for manufacturing verification and mass production [1]. For factories, WAT is an important method of yield monitoring, and for designers, WAT provides useful information to verify process parameters. To this extent, WAT can also be regarded as the basis for process adjustment in the semiconductor manufacturing.

Following WAT, a Circuit Probe (CP) testing system is used to acquire yield data. A test program is provided by engineers for a particular product. The test program describes a test flow including multiple test items, optimally arranged to reduce CP testing time. An ATE then follows the predefined test flow to sequentially probe all dies on a wafer to determine whether a die is good or bad. According to the test results, diagnosis is finally performed for failing wafers with the purpose of identifying root causes.

The main contribution of this work is to propose a learning-based framework for failure prediction at cell level from WAT data using various techniques for feature selection and handling imbalanced classes, as well as finding the optimal combination of these common techniques for performing failure prediction of a given cell. The paper is organized as follows. First, we describe the mathematical framework and then in the experimental evaluation, industrial data and preliminary results are discussed. Finally, future perspectives and directions are highlighted.

II. Prior Work

The prediction of wafer yield based on WAT parameters has been successfully employed to reduce production costs spent on the CP process [2]–[7]. Fig. 1 illustrates a typical procedure of production and quality control for the wafer fabrication. At each stage, a large amount of data can be generated from the corresponding test equipment. Nevertheless, the relationship between WAT and subsequent diagnosis reports has not been sufficiently explored yet. Some obstacles, such as feature

*: These authors contributed equally to this work.
†: Corresponding authors.

978-1-6654-2480-6/21/$31.00 ©2021 IEEE

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 18:25:50 UTC from IEEE Xplore. Restrictions apply.

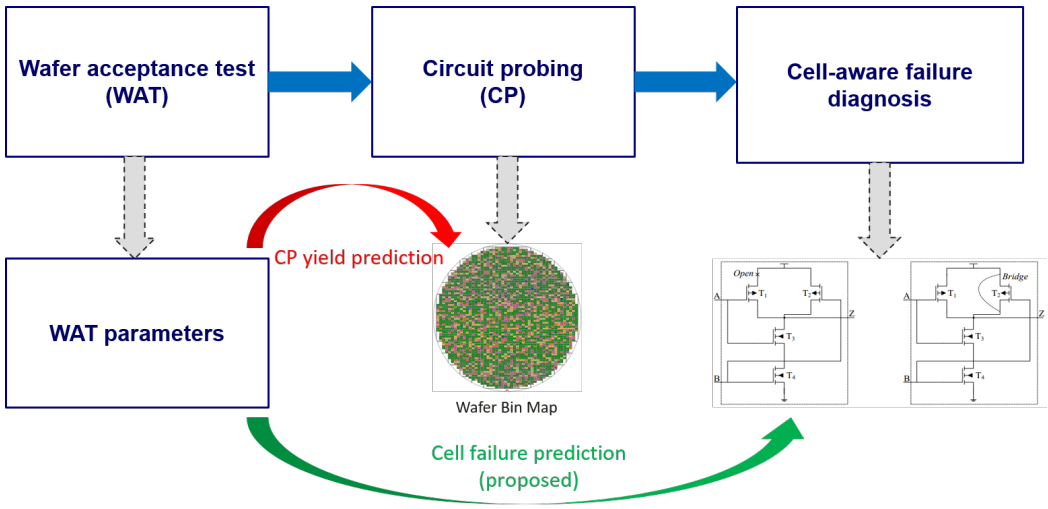


Fig. 1. Illustration of production and quality control of wafers with data generated at each stage. Most existing methods consist of learning the CP yield from WAT data (red arrow). In this paper, we propose to explore the possibility of predicting cell failures based on WAT data (green arrow).

redundancy and label imbalance need to be addressed, as well as lack of available data. This may to some extent explain why there are few studies on the relationship between WAT and diagnosis.

III. Methodology

This section describes the proposed learning framework for cell failure prediction based on WAT data.

In this work, our data samples include both WAT parameters (feature variables) and diagnostic labels (pass or fail, obtained from diagnosis reports) for a given cell. Overall, the data collected during WAT and fault diagnosis would be faced with two major problems: feature redundancy and class imbalance. Feature redundancy occurs due to the fact that WAT includes numerous testing items that might be correlated with one another. Class imbalance is mainly due to the fact that only a small portion of dies are defective in most cases, which implies that common algorithms would result in blind classifiers that tend to predict each new instance as the majority class. To tackle these issues, feature selection and imbalance revision should be considered. In fact, multiple candidate algorithms for feature selection and imbalance revision are available (see Fig. 2). During classification, we propose to evaluate different approaches combining various stages for feature selection, imbalance revision and classification, so as to find the best predictive model for cell failure prediction.

As shown in Fig. 2, for determining the best combination of candidate algorithms, data samples can be split into two sets, namely the training and test sets. The former is used to train multiple candidate classifiers and the latter is used for model evaluation. During the training stage, we also use the K-fold cross-validation and grid search for determining model hyperparameters (e.g., model architecture, regularization, etc.).

A. Feature Selection

Among various algorithms in the literature, three common methods are adopted for feature selection.

1) Principal Component Analysis (PCA): PCA [8] is a common technique for dimensionality reduction, which projects the original high-dimensional data into their lower-dimensional representations so that as much information as possible is preserved about the original data. Thus, the transformed principal components serve as the selected features.

2) Chi-square (Chi2): A chi-square test is used in statistics to evaluate the independence of two events and thus can be used to select those features that have the strongest relationship with the output variable [9]. Features that are highly dependent on the target label shall be selected.

3) Mutual Information (MI): MI is a measure between two random variables X and Y [10], which is a way to represent information of one random variable obtained through the other one. Usually, MI can be calculated as follows:

$$I(X;Y) = \int_X \int_Y \rho(x,y) \log \left[\frac{p(x,y)}{p(x)p(y)} \right] dxdy, \quad (1)$$

where X and Y could be of multiple dimensions. If they are completely unrelated, this integral is equal to zero. Here, feature selection aims to find a subset of features that maximize MI between the selected features and target variable.

B. Imbalance Revision

For handling the problem of class imbalance, two types of techniques are employed. It must be borne in mind that these techniques should only be done in the training stage.

1) Synthetic Minority Over-sampling TEchnique (SMOTE): The over-sampling increases the size of the minority set by duplicating existing observations or generating artificial ones. A widely used method is SMOTE [11], which generates synthetic data based on the selected observation and its neighbors. Moreover, mixed sampling is a combination of under-sampling and over-sampling. Some methods applying the mixed sampling technique are SMOTE+ENN [12] combining SMOTE and Edited Nearest Neighbors (ENN),

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 18:25:50 UTC from IEEE Xplore. Restrictions apply.

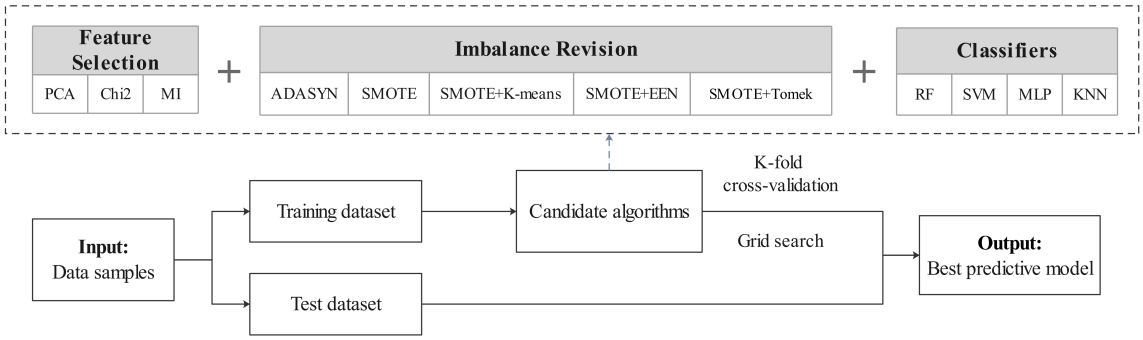


Fig. 2. Proposed framework for optimal model selection. Data samples include both WAT parameters and diagnostic labels. The candidate algorithms for feature selection are PCA, Chi2 and MI. The candidate algorithms for imbalance revision are ADASYN, SMOTE, SMOTE+K-means, SMOTE+ENN and SMOTE+Tomek. The candidate classifiers are RF, SVM, MLP and KNN. See text for details.

SMOTE+Tomek [13] combining SMOTE and Tomek links, and SMOTE+K-means [14] combining SMOTE and K-means.

2) Adaptive Synthetic Sampling (ADASYN): ADASYN [15] refers to an algorithm that generates synthetic data for the minority class. It uses a weighted distribution for reducing bias in the synthetic samples.

C. Classification Algorithms

In this subsection, four popular algorithms are presented for classification tasks.

Algorithm 1 Random forest for classification

Input: Training dataset.
Output: An ensemble of trees and the predicted class.

1: **for** $b \in \{1, \ldots, B\}$ **do**
2: Draw a bootstrap sample from the training set.
3: Develop a decision tree T_b to the bootstrapped data.
4: **end for**
5: Output the ensemble of trees $\{T_b\}_{b=1}^{B}$.
6: To make a prediction at a new point x: Let $\tilde{C}_b(x)$ be the class prediction of the b-th random-forest tree. Then, the predicted class equals to the majority vote of $\{\tilde{C}_b(x)\}_{b=1}^{B}$.

1) Random Forest (RF): RF [16] refers to a modification of bagging based on a large collection of decision trees. The main steps of RF are described in Algorithm (1).

2) Supoort Vector Machine (SVM): The original SVM algorithm was proposed in the 1960s and kernel trick was later introduced to make it more practical [9]. Given a training set $\{x_k, y_k\}_{k=1}^{N}$, where $x_k \in \mathbb{R}^n$ are input patterns and $y_k \in \{-1, +1\}$ are class labels. The classifier can be described as $y(x) = sign\left[w^T \varphi(x) + b\right]$, where $\varphi(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^{n_f}$ mapping arguments from original space to feature space. w is the weight vector and b is the bias. Thus, we have an optimization problem (maximizing the margin):

$$\min_{w,\xi} J(w, \xi) = \min_{w,\xi} \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k, \tag{2}$$

subject to $y_k \left[w^T \varphi\left(x_k\right) + b\right] \geq 1 - \xi_k$ and $\xi_k \geq 0$, where c is a regularization parameter and ξ_k denote slack variables making the classifier more generalized. The primal problem is to estimate w. However, it is usually solved in the dual space by estimating α, namely

$$\min_{\alpha} Q(\alpha) = \min_{w,\xi} \frac{1}{2} \sum_{k,l=1}^{N} y_k y_l K(x_k, x_l) \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k, \tag{3}$$

such that $\sum_{k=1}^{N} \alpha_k y_k = 0$ and $0 \leq \alpha_k \leq c$.

3) Multi-Layer Perceptron (MLP): MLP belongs to a class of feed-forward artificial neural networks (ANN) [17]. In our case, we use a two-hidden-layer MLP that can be written as $y = \phi^{(3)}(\phi^{(2)}(\phi^{(1)}(x; W^{(1)}, b^{(1)}); W^{(2)}, b^{(2)}); W^{(3)}, b^{(3)})$, where W is the weight matrix and b the bias vector. Both W and b can be estimated using the backpropagation algorithm.

4) K-Nearest Neighbors (KNN): KNN is a typical instance-based learning algorithm [18]. The basic idea is that the label of a new query is estimated from the known values of the nearest K training instances. It can be explained as follows:

1) Find K examples closest to the test instance.
2) Output the majority class of these neighbors.

IV. Experiment Evaluation

A. Data Preparation

The experimental evaluation is based on a set of industrial manufacturing data. The data consists of a total of 166 wafer lots, whereas for each lot, only 1 or 2 wafers have undergone the cell-aware diagnosis, which only provides 168 wafers available for classification. In addition, each wafer has 196 dies and WAT parameters are measured on 9 wafer sites. In our case, WAT parameters are mainly composed of linear region threshold voltage (Vtl), saturation region threshold voltage (Vts), leakage current under device shutdown (Ioff), saturation current (Isat), power off (Poff). Vtl and Vts are related to the I/V characteristics of Metal-Oxide-Semiconductor (MOS). Ioff is mainly related to the turn-off current of the device and measures its stability and Poff mainly measures the power consumption when the device is off.

After eliminating the effects of outliers (replaced with the median), 1512 instances including WAT data and diagnostic

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 18:25:50 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SELECTED CLASSIFICATION RESULTS FOR A GIVEN CELL (CASE 1).

Classifier	Feature selection	Imbalance revision	TN	FP	FN	TP	Accuracy	F1	AUC-ROC
RF	None	None	252	2	46	1	0.838	0.069	0.514
	PCA	SMOTE+ENN	116	4	3	248	0.979	0.984	0.974
	MI	SMOTE+ENN	140	10	6	239	0.959	0.967	0.954
SVM	None	None	250	3	46	2	0.836	0.083	0.521
	PCA	SMOTE+ENN	**116**	**3**	**0**	**252**	**0.989**	**0.992**	**0.984**
	Chi2	SMOTE+ENN	51	6	0	258	0.977	0.986	0.942
MLP	None	None	247	8	42	5	0.834	0.256	0.545
	PCA	SMOTE+ENN	101	10	2	252	0.965	0.975	0.948
	MI	SMOTE+ENN	134	16	8	240	0.937	0.950	0.929
KNN	None	None	247	10	36	9	0.845	0.271	0.578
	MI	SMOTE+ENN	139	9	0	251	0.976	0.981	0.967
	PCA	SMOTE+ENN	95	16	0	258	0.956	0.969	0.927

TABLE II
SELECTED CLASSIFICATION RESULTS FOR A GIVEN CELL (CASE 2).

Classifier	Feature selection	Imbalance revision	TN	FP	FN	TP	Accuracy	F1	AUC-ROC
RF	None	None	274	0	28	0	0.906	0.0	0.499
	Chi2	SMOTE+Tomek	**264**	**6**	**6**	**263**	**0.977**	**0.977**	**0.977**
	None	ADASYN	264	7	5	260	0.976	0.976	0.976
SVM	None	None	271	0	31	0	0.896	0.0	0.5
	PCA	SMOTE+ENN	83	6	0	270	0.981	0.987	0.963
	PCA	SMOTE+Tomek	246	18	0	275	0.966	0.968	0.965
MLP	None	None	269	1	31	0	0.890	0.01	0.500
	Chi2	SMOTE+Tomek	234	33	0	272	0.937	0.942	0.937
	None	SMOTE	231	39	1	267	0.924	0.929	0.924
KNN	None	None	271	0	30	0	0.897	0.013	0.502
	MI	SMOTE+ENN	77	27	0	275	0.928	0.953	0.870
	PCA	SMOTE+ENN	59	25	0	271	0.929	0.955	0.850

labels are obtained for a given cell, and each instance corresponds to a wafer site. Here, WAT parameters serve as the feature variables for classification and the diagnostic label is the class label. It should be recalled that we only consider the failure prediction of a given cell, which means that the class label only contains two possible values, namely pass or fail.

For each wafer site, there are a total of 206 WAT parameters and their values lie within different ranges. Therefore, the min-max normalization is adopted for feature scaling, namely $x' = \frac{x-\min(x)}{\max(x)-\min(x)}$, where x denotes the original value and x' is the normalized value.

B. Model Training

This subsection briefly introduces the training setting for our experiments. As shown in Fig. 2, each trial is a combination of feature selection, imbalance revision and classification.

Since there are 4 options for feature selection (i.e., None, PCA, Chi2 and MI), 6 options for imbalance revision (i.e., None, ADASYN, SMOTE, SMOTE+K-means, SMOTE+EEN and SMOTE+Tomek) and 4 options for classifiers (i.e., RF, SVM, MLP and KNN), the total number of trials is 96. For each trial, model parameters and hyperparameters are determined by means of K-fold cross-validation ($K = 10$) and grid search so as to improve classification performance and avoid over-fitting. It should be noted that for each trial, we repeatedly run the pipeline from train-test data split to classification evaluation for 5 times, and the performance metrics are thus averaged.

C. Results and Discussion

This subsection presents some selected results of the proposed approach, as well as the influence of feature dimension on classification performance.

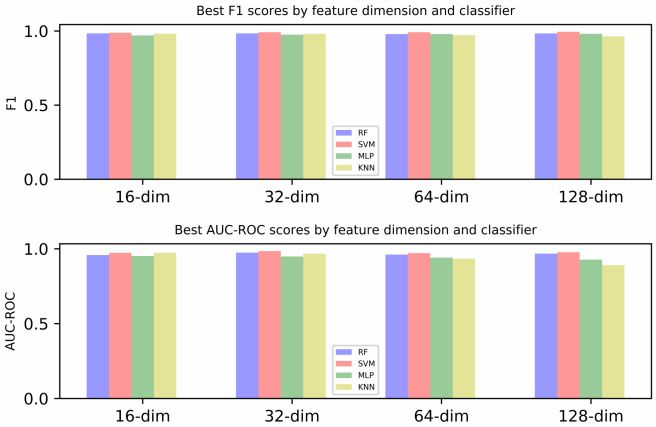


Fig. 3. Model performance as a function of reduced feature dimension.

1) Model Comparison: For each classifier, there are 24 combinations of feature selection and imbalance revision methods in total, and the metrics for performance evaluation are accuracy, F1 and Area Under the Receiver Operating Characteristic curve (AUC-ROC), True Negative (TN), False Positive (FP), False Negative (FN) and True Positive (TP). The accuracy score can be calculated with $accuracy = (TP + TN)/(TP + TN + FP + FN)$. The F1 score can be calculated with $F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$ where $Recall = TP/(TP + FN)$ and $Precision = TP/(TP + FP)$. A Receiver Operating Characteristic (ROC) curve shows the trade-off between sensitivity and specificity of a classifier. The F1 and AUC-ROC scores are often used to measure the overall performance of a binary classifier, especially in the case of imbalanced classes. The closer F1 or AUC-ROC for a model comes to 1, the better it is [9].

For each classifier, we selected the case with neither feature selection nor imbalance revision as well as other two cases with highest F1 or AUC-ROC scores.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 18:25:50 UTC from IEEE Xplore. Restrictions apply.

TABLE III
MEAN VALUES (NORMALIZED) OF THE SELECTED WAT PARAMETERS (FAIL/PASS GROUP) FOR FAILURE PREDICTION OF A GIVEN CELL.

Parameter	Mean (FAIL)	Mean (PASS)
Vtl1	0.530	0.400
Vtl2	0.501	0.410
Vts1	0.523	0.491

From Tables I and II (classification results for two selected cells), the following conclusions can be drawn:

- After applying feature selection and imbalance revision, in the best-case scenario, both F1 and AUC-ROC scores can reach more than 95%. Comparing to the single classifier, the proposed framework shows great advantages.
- Regarding feature selection, it is shown that PCA and Chi2 generally achieve a better performance. As for imbalance revision, SMOTE+ENN seems to be better and outperforms other options.

It should also be noted that the proposed framework can automatically give us the optimal combined model for cell failure prediction, and for different cells, the best combination of candidate algorithms might vary.

2) Impact of feature dimension on classification performance: When using feature selection, no matter which method (PCA, Chi2 or MI) is adopted, it is required to determine the reduced dimension or the number of selected features. Hence, the number of feature variables would have an impact on the model performance. The present work examines four reduced feature dimensions: 16, 32, 64 and 128. Fig. 3 shows the best F1 and AUC-ROC scores of each classifier among 24 combinations of feature selection and imbalance revision algorithms under different reduced dimensions.

From Fig. 3, it can be seen that under each reduced dimension and classifier, the model performance varies without significance and overall SVM outperforms other classifiers. However, the best AUC-ROC score shows an obvious drop when the reduced dimension is set to 128, which is likely caused by over-fitting due to the model complexity.

3) Discussion: Based on the above results, we would like to illustrate an application scenario of the proposed method. Taking a cell "AOI" for example, it would be possible to identify the most influential parameters in predicting the failure of this cell.

By jointly considering the cell-aware diagnosis results, three selected parameters are filtered out using Chi2 feature selection and listed in Table III. Here, the parameters Vts and Vtl correspond to the transistors with gate length of 11nm. In Table III, it is found that the mean values (normalized) of the pass group are systematically smaller than those of the fail group. Therefore, during the manufacturing process, if the measured values of Vts and Vtl are too large, it could indicate that the corresponding cell might fail during diagnosis and further investigation of failure mechanisms should be carried out accordingly.

V. CONCLUSION AND PERSPECTIVES

In this paper, a machine learning-based approach is presented for failure prediction at cell level based on WAT parameters. The proposed methodology relies on binary classification combined with feature selection and imbalance revision techniques. It has been demonstrated that using the proposed framework, we are able to obtain both high accuracy and AUC-ROC scores for a set of industrial manufacturing data. Future research directions include verifying generalization capabilities in different designs and taking into account cell-internal features for improving predictive power.

REFERENCES

[1] J. Ruth and R. Berndt, "Quality control for ultrafiltration of ultrapure water production for high end semiconductor manufacturing," in *2016 27th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*, 2016, pp. 16–22.

[2] F. Tan, T. Pan, Z. Li, and S. Chen, "Survey on run-to-run control algorithms in high-mix semiconductor manufacturing processes," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 6, pp. 1435–1444, 2015.

[3] J. Moyne, J. Samantaray, and M. Armacost, "Big data capabilities applied to semiconductor manufacturing advanced process control," *IEEE Transactions on Semiconductor Manufacturing*, vol. 29, no. 4, pp. 283–291, 2016.

[4] J. Moyne and J. Iskandar, "Big data analytics for smart manufacturing: Case studies in semiconductor manufacturing," *Processes*, vol. 5, no. 3, 2017. [Online]. Available: https://www.mdpi.com/2227-9717/5/3/39

[5] Q. Huang, C. Fang, S. Mittal, and R. D. Blanton, "Improving diagnosis efficiency via machine learning," in *2018 IEEE International Test Conference (ITC)*. IEEE, 2018, pp. 1–10.

[6] H.-G. Stratigopoulos, "Machine learning applications in ic testing," in *2018 IEEE 23rd European Test Symposium (ETS)*. IEEE, 2018, pp. 1–10.

[7] H. Xu, J. Zhang, Y. Lv, and P. Zheng, "Hybrid feature selection for wafer acceptance test parameters in semiconductor manufacturing," *IEEE Access*, vol. 8, pp. 17 320–17 330, 2020.

[8] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[10] B. C. Ross, "Mutual information between discrete and continuous data sets," *PloS one*, vol. 9, no. 2, p. e87357, 2014.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[12] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[13] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: a case study." in *WOB*, 2003, pp. 10–18.

[14] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and smote," *Information Sciences*, vol. 465, pp. 1 – 20, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518304997

[15] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1322–1328.

[16] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[17] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[18] O. Kramer, "K-nearest neighbors," in *Dimensionality reduction with unsupervised nearest neighbors*. Springer, 2013, pp. 13–23.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 18:25:50 UTC from IEEE Xplore. Restrictions apply.

Fault Diagnosis of Wafer Acceptance Test and Chip Probing Between Front-End-of-Line and Back-End-of-Line Processes

Shu-Kai S. Fan, Chun-Wei Cheng, and Du-Ming Tsai

***Abstract*— With the rapid development of the semiconductor industry, fault diagnosis is an important task in routine operations to determine the root cause for faults that occur. A tool in manufacturing processes is equipped with a wide variety of sensors that record different types of process data. Wafers being processed are accompanied by a considerable amount of process data as a multivariate time series. Identifying the key process parameters and the corresponding process operations where faults occur can be used to facilitate the tasks of monitoring the process, maintaining the stability of the process, and stabilizing wafer production yield. This article proposes a novel solution procedure for fault diagnosis of wafer acceptance test (WAT) and chip probing (CP) using machine learning (ML). Based on the process flow of wafers and the corresponding process data, a sampling method, called synthetic minority oversampling technique (SMOTE), is first used to augment classification models with an imbalanced process dataset, and the best-practice SMOTE ratio is sought by using four competitive classifiers. By means of principal component analysis (PCA), the original data are transformed into visualizations to explore the distributions of good and bad lots of wafers. Based on the comparison of the four classifiers used in the test, the proposed logistic regression with data augmentation (LR-SMOTE) performs best. The ten most important features identified by using the LR are collected to determine potential failure operations. The identified failures in operations returned by using the proposed solution procedure for fault diagnosis of WAT and CP were confirmed by the engineers who work in the domain.**

***Note to Practitioners*—In semiconductor manufacturing practice, during the WAT and CP, domain engineers need to trace upstream the key process parameters and the corresponding key operations in the front-end-of-line (FEOL) process if any faults are detected in the back-end-of-line (BEOL) process. In fact, the FEOL and the BEOL are located in completely separate plants with different clean room standards. Bridging the information gap between these two different processes merits thorough scrutiny. Another challenge posed by fault diagnosis of WAT and CP is that the total number of features (i.e., process parameters) involved in the FEOL can be tens of thousands, leading to an adverse effect of data sparsity on model building.**

***Index Terms*— Fault diagnosis, feature engineering, machine learning (ML), semiconductor manufacturing.**

Manuscript received 1 June 2021; revised 25 July 2021; accepted 16 August 2021. Date of publication 31 August 2021; date of current version 13 October 2022. This article was recommended for publication by Associate Editor L. Moench and Editor F.-T. Cheng upon evaluation of the reviewers' comments. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-027-028-MY3. *(Corresponding author: Shu-Kai S. Fan.)*

Shu-Kai S. Fan and Chun-Wei Cheng are with the Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei 10608, Taiwan (e-mail: morrisfan@ntut.edu.tw).

Du-Ming Tsai is with the Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan 32003, Taiwan.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2021.3106011.

Digital Object Identifier 10.1109/TASE.2021.3106011

I. Introduction

RAMPING up yield and maintaining quality in semiconductor manufacturing is critical as the feature size of devices continuously shrinks. Effective identification and diagnosis of potential failures in operations from the mass volume of process data pose a great challenge for manufacturers [1]. To date, effective data mining tools offered by engineering data analysis (EDA) have provided a solid basis for well-tuned manufacturing processes. However, conducting fault diagnosis is time-consuming and tedious in a multiple-stage process flow in wafer fabrication. Therefore, a systematic procedure for fault diagnosis warrants a full investigation.

Machine learning (ML) has emerged as a new discipline to facilitate rapid data analysis [2]. It is a collection of methods for automatically extracting latent information and rules from large amounts of data. The success of the ML algorithms, such as the logistic regression algorithm [3], [4] that plays a central role in this article, depends largely on the "quality" of datasets that are used to develop classifier learning. Most ML algorithms require the use of similar numbers of instances for each of the classes. However, balanced datasets are not always possible in practice. The incorrect classification of instances of the minority class usually costs significantly more than the incorrect classification of those of the majority class. Minority class instances are rare but always related to a critical issue that should be solved urgently. Hence, the problem of how to handle imbalanced datasets in ML has received considerable attention in the recent literature [5]–[9].

The primary purpose of this article is to develop a data-driven solution procedure for fault diagnosis of wafer acceptance test (WAT) [10]–[12] and chip probing (CP) [13], [14] in semiconductor manufacturing. These two processes will be discussed briefly in Section II. We refer interested readers to the following works about fault detection (FD) and diagnosis using ML algorithms in semiconductor manufacturing [1], [15]–[18].

II. Motivation and Problem Statement

In current semiconductor manufacturing practice, a large amount of process data is generated, which is collected from

1545-5955 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

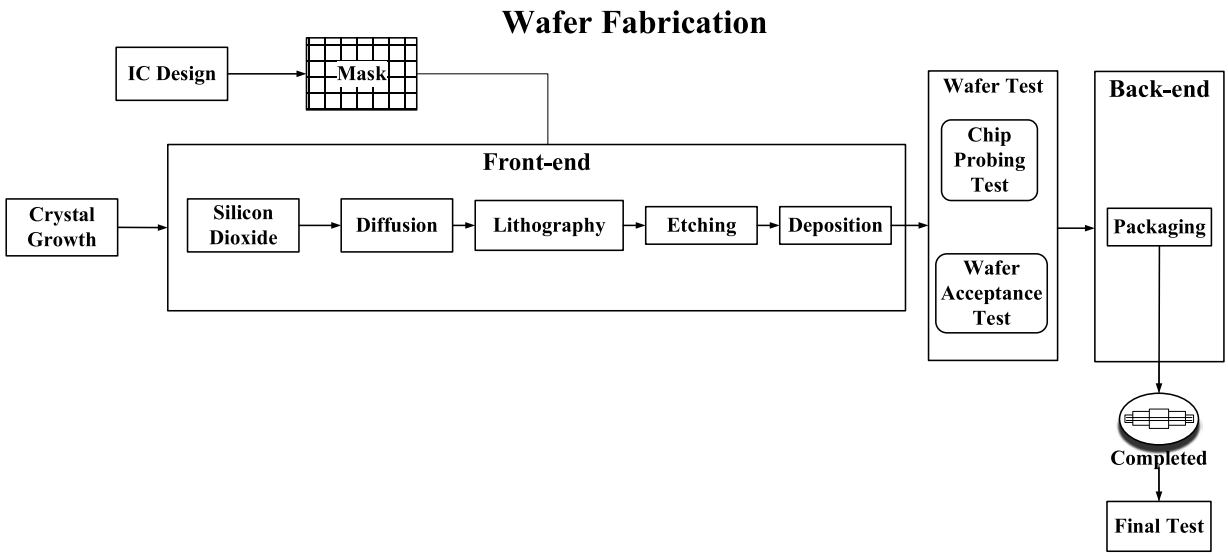


Fig. 1. Typical wafer end-of-life process flow.

different types of sensors installed on a sequence of process equipment, such as thin-film deposition, implantation, photolithography, etching, testing, and probing. Data collection covers every facet of important information that can be used to observe the production status. The collected process data are used for two purposes: to provide real-time data for FD, and to provide online data for *post hoc* analysis. The fault diagnosis of WAT and CP addressed in this article is of the latter type. If a fault is detected in testing, potential failures in operations need to be traced back to the original process data during wafer fabrication. However, manipulating such large and complex multiple-step data in connection with WAT and CP represents a serious challenge for fault diagnosis. In the following sections, wafer fabrication, WAT, and CP are elaborated.

A. Semiconductor Manufacturing

Semiconductor chips are made of integrated circuits (ICs) with many components, such as transistors, capacitors, resistors, and wiring formed on the surface of a wafer. Electronic chips are made of 3-D structures, usually consisting of many complex circuit layers and thousands of electronic devices on a tiny area [19]. Semiconductor manufacturing processes take place in a cleanroom where temperature, atmospheric pressure, and humidity are strictly controlled to avoid contamination. A typical manufacturing process involves hundreds of process steps, layer by layer, on a bare wafer that contains hundreds of chips.

Generally, semiconductor manufacturing can be divided into wafer fabrication, wafer probe, packaging, and final test [20], as shown in Fig. 1. Wafer fabrication is referred to as the front-end-of-line (FEOL) process. Assembly, including packaging and final test, is referred to as the back-end-of-line (BEOL) process. The data source for this article is wafer fabrication. Circumscribed by solid lines on the left are major repetitive unit processing steps of wafer fabrication, including thin film, photolithography, etching, ion implantation, diffusion, and chemical mechanical planarization (CMP). From the process integration point of view, statistical stability of the process at the in-line level does not necessarily guarantee that of the end-of-line parametric data [21]. First, not all the important process parameters are clearly identified, and not all the dependencies among process parameters are taken into consideration in the in-line process control. Therefore, a systematic approach to fault diagnosis of semiconductor manufacturing processes based on the end-of-line performance outcome (i.e., regular or defective) is needed. According to the wafer flow in Fig. 1, semiconductor manufacturing data can be roughly divided into two stages: processing and device testing.

B. Wafer Acceptance Test

WAT is an essential testing process of semiconductor fabrication prior to shipping to assembly. The collected testing parameters can be used to identify wafer defects, improve product yield, and control production costs [11], [12]. In order to ensure wafer quality, several electrical characteristics of different devices under test (DUT) are measured. To evaluate the quality and stability of the manufacturing process, WAT is performed against the test key at the end of the FEOL process where every DUT in the monitoring test lines is measured. To circumvent potential damage to the wafer, the monitoring test lines are placed on the scribe lines of a wafer.

While collecting WAT data, a variety of test structures are designed to impose upon the isles between chips in a wafer at the initial stage of wafer fabrication, in an attempt to examine the electrical properties/performances of the DUTs fabricated without damaging the circuitry in chips. The measured data of electrical characterization parameters are collectively called the WAT data. Analysis of the WAT data provides the process engineers with a rapid and direct assessment of electrical problems that can be attributed to process difficulties. It serves as a critical quality checkpoint for detecting the abnormality of manufacturing processes and estimating the characteristics of various electronic devices on the same wafer [11], [12].

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

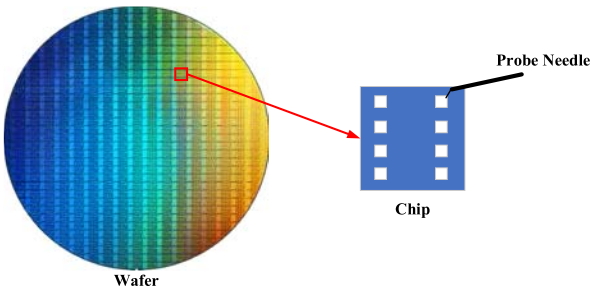


Fig. 2. Relationship between the probe needle, the chip, and the wafer.

C. Chip Probing

CP is used to identify the defective dies against the designed electrical specification in a wafer form and also so-called wafer sorting. CP is performed on every wafer- and die-basis. If not, defective dies will go through the pipeline to assembly, resulting in unnecessary expenses in packaging. CP can greatly reduce the packaging and testing costs and the wafer yield can be maintained.

The method of wafer needle testing uses an extremely sophisticated probe having contact with the wafer through the probe interface card, an interface between an electronic test system and a wafer. In general, the probe card is mechanically docked to a prober and electrically connected to a tester. The test procedure requires the signal from the tester into the grain, and receives the corresponding grain output signal and then transmits it back to the tester. The outcome of the test procedure determines whether the wafer function meets the design specifications or not. If all test patterns pass for a particular die, its position on a wafer is marked for later use during the packaging stage. The relationship is shown in Fig. 2.

The first challenge to be faced during WAT and CP is that the domain engineers need to trace upstream the key process parameters and the corresponding key operations in the FEOL process if any faults are detected in the BEOL process. In fact, the FEOL and BEOL are located in completely separate plants with different cleanroom standards. Bridging the information gap between the two processes warrants a full investigation. The second challenge posed by fault diagnosis of WAT and CP is that the total number of features (i.e., process parameters) involved in FEOL can be tens of thousands, and that can have an adverse effect of data sparsity on model building.

III. Proposed Solution Procedure for Fault Diagnosis of WAT and CP Data

A. Data Representation

The dataset was collected from an 8-in wafer fabrication process, recorded by the sensors during the production of liquid crystal display (LCD) controller ICs provided by a globally well-known semiconductor fabrication plant in Hsinchu Science Park, Hsinchu, Taiwan. In most semiconductor manufacturing processes, a variety of sensors are installed on the process equipment to record the values of every process parameter as time progresses. During the processing period, the sensor readings reflect the status and condition of a wafer over time. This is also known as raw trace data. The investigation of the raw trace data into the fault diagnosis in this study is constrained within the context of WAT and CP, instead of FD during wafer fabrication.

In order to conduct data analytics with a reasonable level of intelligence, we intend to develop a generalized solution procedure for dealing with the raw trace dataset that accommodates fault diagnoses of WAT and CP. However, the root causes of faults in these two scenarios and the multiple-step process sequence could be different. For this reason, we wish to design a more flexible approach to handle both scenarios simultaneously, so a set of ML approaches are selected to build the solution procedure.

For the WAT data, the front and back ends of the line of wafer production are shown in Fig. 3. The studied wafer dataset includes process/equipment (EQP) data from process stations, test stations, and measurement stations taken at specific processing steps. Process data reflect the condition of the equipment; the WAT and CP test data can eventually be affected by any condition or status changes in the equipment. According to the internal report provided by the advanced process control (APC) team of the semiconductor fabrication plant under investigation, the bad lot of the WAT data, detected by the first-cut diagnosis, is caused by an abnormal process shift as shown in the inner loop of Fig. 3. For the CP data, the same dataset as in WAT is used. The only difference between the WAT and CP data is in the multiple-step process sequence. The bad lot of the CP data, detected by the second-cut diagnosis, is due to the yield loss. The reason for low yield is primarily attributed to uneven grinding during the CMP process, resulting in residual tungsten and short circuits.

B. Proposed Solution Procedure for Fault Diagnosis

Data cleaning is extremely critical for the success of ML applications in the industry. To make the analysis of fault diagnosis more trustworthy, we explored the dataset and removed incomplete or incorrectly formatted data. Several python codes were done to browse the datasets (in CSV format). To deal with such a large pool of datasets, a list of missing data percentage for each feature was listed to check missing values. To detect unnecessary data, a list of features with a high percentage of the same value was created. To detect inconsistent data format, the feature was converted from string to date–time format. While combining related data files, the avoidance of data duplication and mislabeling was carried out carefully. In addition, a minimum threshold is set to remove uninfluential features with minuscule variability; that is, they are really close to constant. Then, only the numerical readings recorded by sensors in the process site were considered in as much as most errors originated from process stations. The goal of this step is to conduct data cleaning, table mapping, and rearrangement. To cope with the imbalance issue of data classes, the number of samples of the bad lot must be augmented to make the classifier learn and predict effectively. In the investigation of the WAT and CP scenarios, the synthetic minority over-sampling technique (SMOTE) [22] is chosen. The main concept of the SMOTE

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

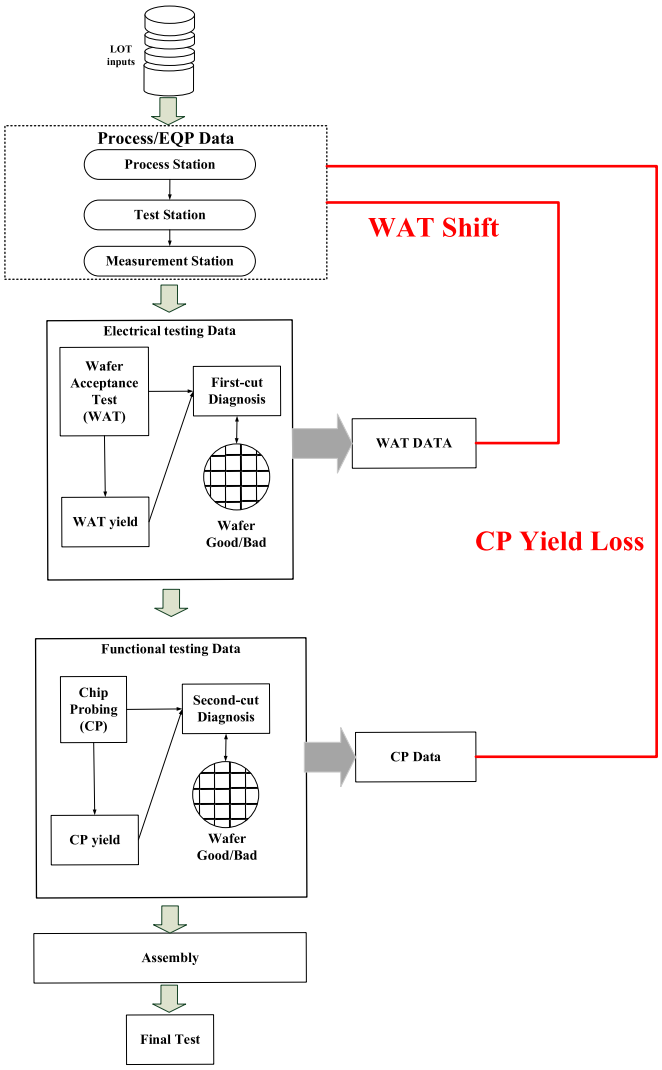


Fig. 3. Production and quality control flowchart of the WAT and CP data.

is to synthesize some samples in close proximity to a few samples. For the visualization purpose, principle component analysis (PCA) [23], [24] is used to compare the data distributions before and after data augmentation. Using a variety of sampling ratios in the SMOTE, different classification models are compared while classifying the augmented dataset for evaluation purposes. Last, the best performed LR-SMOTE classifier is then used to execute feature selection for the key parameter identification. The identified key parameters are clustered according to their corresponding operations. In terms of the preceding steps, failure in operations can be pinpointed by using a set of ML methods and a systematic fault diagnosis system of WAT and CP can be built. The framework of the proposed solution procedure is illustrated in Fig. 4. The primary purpose of classification models is to identify feature importance during classification.

C. Synthetic Minority Over-Sampling Technique

To deal with the imbalance problem of the WAT data and CP data in semiconductor practice, the number of samples

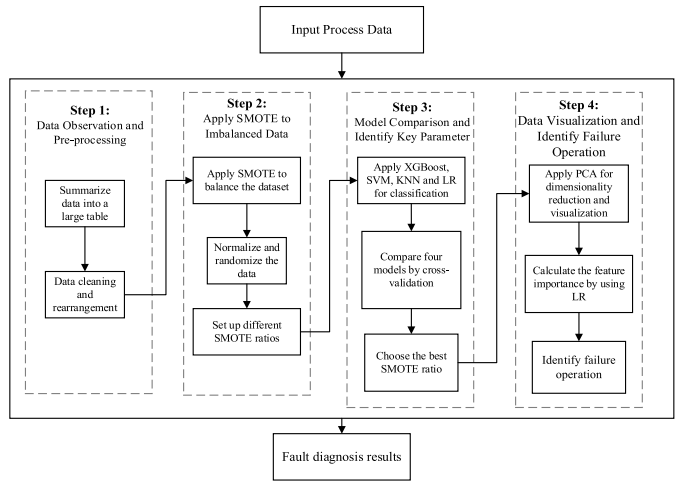


Fig. 4. Proposed solution procedure of fault diagnosis.

of the bad lot must be increased. In the literature, various studies [6]–[9] have shown that resampling data is of primary importance to mitigate this problem. In this article, the SMOTE is used. The SMOTE is a powerful sampling method that has been successful in various applications. The SMOTE creates synthesized data based on the feature space similarities between existing minority examples.

To be specific, for subset $S_{\min} \in S$ where S denotes the total training samples of all classes and $S_{\min}$ denotes the subset that contains the samples of a minority class, consider the K-nearest neighbors for each sample in the WAT data and the CP data, and $\boldsymbol{x}_i \in S_{\min}$ in the feature space $\mathbb{R}^n$, for some specified integer K. The K-nearest neighbors are defined as the K elements of $S_{\min}$ whose Euclidian distance between them and $\boldsymbol{x}_i$ under consideration exhibits the smallest magnitude in $\mathbb{R}^n$. To create a synthetic sample, one of the K-nearest neighbors is randomly selected, then multiplying the corresponding feature vector difference via a random number between [0,1] and adding this vector to $\boldsymbol{x}_i$ gives

$$\boldsymbol{x}_{\text{new}} = \boldsymbol{x}_i + (\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i) \cdot \delta \tag{1}$$

where $\boldsymbol{x}_i \in S_{\min}$ is the minority instance under consideration, $\hat{\boldsymbol{x}}_i$ is one of the K-nearest neighbors for $\boldsymbol{x}_i$ where $\hat{\boldsymbol{x}}_i \in S_{\min}$, and $\delta \in [0, 1]$ is a random number. The resulting synthetic instance according to (1) is a vector point along the line segment joining $\boldsymbol{x}_i$ under consideration and the randomly selected K-nearest neighbor $\hat{\boldsymbol{x}}_i$. The Algorithm of SMOTE is formalized as follows.

Algorithm SMOTE $(S_{\min}, k)$

1. Given the set of a minority class, $S_{\min}$; the number of nearest neighbors, k;
2. Specify a random sample $\boldsymbol{x}_i \in S_{\min}$;
3. Seek k nearest neighbors of $\boldsymbol{x}_i$;
4. Randomly choose one of k nearest neighbors, $\hat{\boldsymbol{x}}_i$;
5. Create the synthetic sample by $\boldsymbol{x}_{new} = \boldsymbol{x}_i + (\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i) \cdot \delta$, where $\delta \in [0, 1]$.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

Through the sample generation by means of the SMOTE, we choose a best-practice augmentation ratio based on the simulation results of the WAT data and the CP data to be demonstrated in Section IV.

D. Data Visualization via PCA

The purpose of data visualization is to create a graphic representation of data. When there are staggeringly large numbers of parameters involved, high-dimensional data will be beyond direct observation. If the high-dimensional data in WAT and CP can be transformed into a low-dimensional space, then data visualization can be utilized to compare the distributions of good and bad lots with and without data augmentation using the SMOTE oversampling method. By doing so, the effectiveness of the SMOTE and its parameter setting to differentiate the good and bad lots can be visually assessed.

PCA has been a mainstay tool in modern data analysis to change the basis of datasets in diverse fields. It is a straightforward and non-parametric method for extracting latent patterns from massive datasets. The objective of PCA is to reduce the dimensionality of the dataset under investigation while retaining as much information as possible of the inherent variability within the data. Principal components are linear transformations of the original dataset, and the components are uncorrelated (i.e., orthogonal) and ordered so that the first few important components contain the most information about the original dataset [24]. The variance retained in the transformed data is maximized, and the least squared reconstruction error is minimized while choosing a projection matrix. Suppose that $\boldsymbol{X}$ indicates an $m \times n$ data matrix containing m samples of an n-dimensional vector $\boldsymbol{x}$ of process parameters selected from the WAT data or the CP data, having the n-dimensional mean vector as given by

$$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i \tag{2}$$

and the covariance matrix $\boldsymbol{\Sigma}$ in the following:

$$\begin{aligned}\boldsymbol{\Sigma} &= \frac{1}{m-1}\sum_{i=1}^{m}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^{\mathrm{T}} \\ &= \left[\sigma_{i,j}\right]_{n\times n} \\ &= \begin{bmatrix} \sigma_{1,1} & \sigma_{1,2} & \cdots & \sigma_{1,n} \\ & \sigma_{2,2} & \cdots & \sigma_{2,n} \\ & & \ddots & \vdots \\ & \text{symm.} & & \sigma_{n,n} \end{bmatrix}\end{aligned} \tag{3}$$

where $\boldsymbol{\Sigma}$ should be a square matrix, and $\sigma_{i,j}$ denotes the covariance between features i and j, for $i, j = 1, 2, \ldots, n$. Because $\boldsymbol{\Sigma}$ is a covariance matrix, it is at least a positive semidefinite (p.s.d.) matrix. All eigenvalues of $\boldsymbol{\Sigma}$ must be greater or equal to zero. That is, the sorted eigenvalues of $\boldsymbol{\Sigma}$ satisfy $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. If $\boldsymbol{\Sigma}$ has the full rank, then $\lambda_1 > \lambda_2 > \cdots > \lambda_n \geq 0$. The orthonormal eigenvector matrix that contains n eigenvectors, $(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n)$ is denoted by $\mathbf{A}$. The Algorithm of PCA for the WAT data and the CP data can be formalized as follows.

Algorithm PCA $(\boldsymbol{X}, c)$ for Selecting Top c Component

1. Given data matrix $\boldsymbol{X}$ of size $m \times n$;
2. Compute the mean vector $\bar{\boldsymbol{x}}$;
3. Compute the covariance matrix $\boldsymbol{\Sigma}$;
4. Compute the eigenvalues of $\boldsymbol{\Sigma}$ and sort the eigenvalues;
5. Compute the matrix $\boldsymbol{A}$;
6. Choose c eigenvectors with the largest eigenvalues to form a $c \times n$ matrix $\boldsymbol{W}$ (where every row represents an eigenvector), and $\boldsymbol{W} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_c]^{\mathrm{T}}$;
7. Transform the sample data by $\boldsymbol{X}\boldsymbol{W}^{\mathrm{T}}$ into the new feature space spanned by the row vectors of $\boldsymbol{W}$.

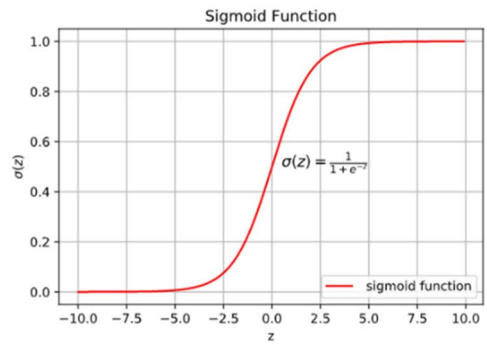


Fig. 5. Sigmoid function.

In a later experiment in Section IV, PCA will be used to examine the best-practice augmentation ratio of the SMOTE used in the WAT data and the CP data to demonstrate the disparity between the distributions of good and bad lots.

E. Feature Selection and Classification in Logistic Regression

In ML, logical regression (LR) is basically an extension of linear regression to solve a classification problem where observations are assigned to a discrete set of classes [25], [26]. In this article, LR is used to classify a lot into one of the good and bad observations. Here, the main idea is to determine the correlation between the input attribute vector $\boldsymbol{x}$ in the WAT data or the CP data and output rank y (good or bad) through probability estimation. The prediction model is constructed using the logistic function shown in (4), i.e., the sigmoid function. The graph of the logistic function is an S-shaped curve, as shown in Fig. 5, with the curve centered at $(0, 0.5)$. The functional value grows rapidly near the center, and gradually approaches the endpoint extremum 0 or 1 as the absolute value of z becomes asymptotically large, as given by

$$\begin{aligned} z &= \beta_0 + \sum_{i=1}^{n} \beta_i x_i \\ y &= \sigma(z) = 1/(1+e^{-z}) \end{aligned} \tag{4}$$

where β_0 is the bias term and β_i's are the unknown weights of a vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$ of process parameters of length n. In (4), $\sigma(z)$ takes any real-valued number and maps it into a legal probability in the range $[0, 1]$. By total law of

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

probability, this assumption can be realized according to

$$p(y=1|\boldsymbol{x}) = \sigma(z)$$
$$p(y=0|\boldsymbol{x}) = 1-\sigma(z). \tag{5}$$

To make a decision, the outcome rank of $y = 1$ (bad lot) should be predicted as $p \geq 0.5$ (i.e., $z \geq 0$); $y = 0$ (good lot) as $p < 0.5$ (i.e., $z < 0$). Equation (5) makes LR a linear classifier. The classification boundary to differentiate the two classes is defined by the solution of $z = 0$. LR is similar to linear regression, but the curve is constructed using the natural logarithm of the "odds" of the outcome rank, rather than a pure probability. The predictors do not have to be normally distributed or have equal variance in each class. To train an LR model using the WAT data or the CP data, there are two steps: 1) write the log-likelihood form of all the training data based on a Bernoulli distribution in (5); and 2) use the numerical optimization techniques to solve the maximum likelihood estimation (MLE) problem, such as the gradient descent algorithm. Alternatively, the iterative weighed least squares algorithm is used to solve the cross-entropy loss function for parameter estimation.

Once classification is accomplished, feature importance needs to be calculated in order to select top features from an entire set of features. The importance of each selected feature accumulates according to its corresponding operation, and then potential failure operations can be identified. The Algorithm of failure operation identification is formalized as follows.

Algorithm Failure Operation Identification $(\boldsymbol{X}, \tau)$ for Selecting Top τ Features to Identify Failure Operations

1. Perform logistic regression on data matrix $\boldsymbol{X}$ of size $m \times n$ with output rank y (good or bad);
2. Determine feature importance by coefficient ranking in logistic regression;
3. Screen top τ features;
4. Assign feature importance scores of the selected τ features to the operations where they belong;
5. Accumulate feature importance scores for every operations and return the one with the maximum accumulated score.

IV. Empirical Study and Validation Results

In this section, the real-data cases are investigated, where a large number of sensor readings in the multiple-step process flow of manufacturing LCD driver ICs are collected to form the WAT and CP data. In every semiconductor manufacturing process, various equipment sensors record the values of each parameter on the tool. As the silicon wafers are processed, the sensor readings can be used to observe and examine temporal changes of every parameter. Hence, if faults occur in wafer testing and probing, the process engineers can inspect the sensor readings carefully to identify potential root causes.

To conduct fault diagnosis in an intelligence-added manner, we introduce a portmanteau of ML techniques to analyze the raw trace data, in which two different cases of fault diagnosis are studied, i.e., in WAT and CP. Both cases involving two different LCD drivers arose from the same fabrication plant, but the root cause of the problem and the sequence of the multiple-step process flow were quite different from each other. The five main data files in the WAT data and the CP data include the WAT workflow, the CP workflow, the process stations, the test stations, and the measurement stations, the last three of which are indicated in Fig. 3. We propose a generic/common solution procedure for both cases. To this end, LR and different supervised ML techniques are tested and compared while choosing an appropriate model for classification and feature importance discrimination. For the WAT data, there are 42 lots in total, ten bad lots and 32 good lots, and 12 915 features from the three stations and one outcome rank (i.e., annotation). For the CP data, there are 75 lots in total, seven bad lots and 68 good lots, and 12 776 features from the three stations and one outcome rank. In communication with the on-site engineers, unrelated information and categorical data were removed, and only the numerical process data from the process stations are taken into account for both cases of fault diagnosis. As shown in Fig. 6, 3023 process parameters in WAT and 2068 process parameters in CP were considered in ML model building. Even though WAT and CP are performed to obtain different test responses, the WAT/CP response required in the proposed fault diagnostic procedure is merely binary, i.e., good or bad. For the purposes of illustration, a few examples of the process parameters are exhibited in Table I. Only numerical parameters A5, D1, T3, and VP in the table are considered in the building of ML models. For reasons of confidentiality, the physical meanings of the process parameters have been de-identified.

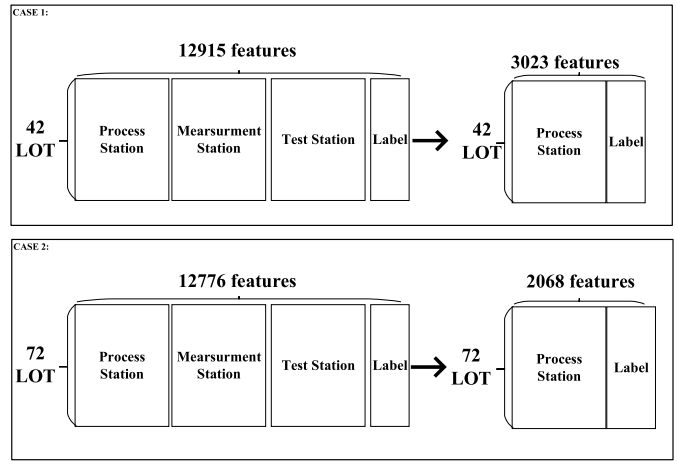


Fig. 6. Data cleansing and arrangement of the WAT and CP data.

Due to the insufficient sample size for the bad lot, we augment the number of samples of the bad lot through SMOTE. In Table II, the imbalance ratios of the WAT data and the CP data are computed by

$$\text{Imbalance Ratio} = 1 - \frac{\alpha_1}{\alpha_2} \tag{6}$$

where α_1 denotes the number of objects in the minority class and α_2 denotes the number of objects in the majority class. The minority (positive) class consists of the data corresponding to the bad lot, while the majority (negative) class consists of the

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PROCESS PARAMETERS

PARAMETER ID	Value	Remark	Station
A5	-350	Boundary Coordinate	Process
D1	-10	Robotic-Arm Coordinate	Process
ID	Standard_No.11	Route Rule	Process
R2	PS01	Tool ID	Process
T3	399.4	Temperature	Process
YR	AALW37	Production Recipe	Process
VP	73	Grinding platform speed	Process

TABLE II
IMBALANCE RATIOS OF THE WAT AND THE CP DATA

Dataset	Total number of bad (positive) lots	Total number of good (negative) lots	Imbalance ratio
WAT data	10	32	0.6875
CP data	7	68	0.8971

data corresponding to the good lot. As shown in the table, the WAT data (68.75%) exhibits less imbalance than the CP data (89.71%).

For each case, the SMOTE minority class samples are generated in different ratios, from 20% of over sampling to 100% of over sampling with respect to the majority class as shown in Table III, for purposes of evaluation. The ratio that performs best among different classifiers is reported. The SMOTE ratio in the table corresponds to the desired ratio of the number of samples in the minority class over the number of samples in the majority class after resampling. Therefore, the oversampling ratio is expressed by

$$\alpha_{\text{os}} = \frac{N_{\text{positive}}}{N_{\text{negative}}} \tag{7}$$

where N_{positive} is the number of samples in the minority class after resampling and N_{negative} is the number of samples in the majority class.

We now proceed to the experiment of lot classification. In this section, LR and three widely accepted classifiers, k-NN, extreme gradient boosting (XGBoost), and support vector machine (SVM) are chosen for the evaluation. All the implementations are executed in the Python package [27]. To provide a fair evaluation, the classification results from ten-fold cross validation are reported. The average accuracy and F_β-score computed over ten folds are used as the yardstick for assessment, as defined by

$$\text{Accuracy} = \frac{1}{10}\sum_{j=1}^{10}\left(\frac{\text{TP}_j + \text{TN}_j}{\text{TP}_j + \text{TN}_j + \text{FP}_j + \text{FN}_j}\right) \tag{8}$$

$$\text{F}_\beta\text{-score} = \frac{1}{10}\sum_{j=1}^{10}\left(\frac{(1+\beta^2)\text{TP}_j}{(1+\beta^2)\text{TP}_j + \text{FP}_j + \beta^2\text{FN}_j}\right) \tag{9}$$

where TP denotes true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative; j is the fold index. For every studied classifier, the original

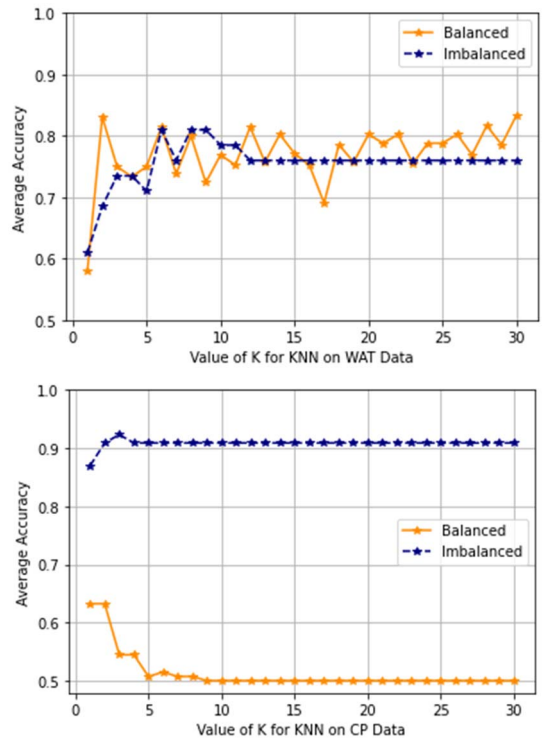


Fig. 7. Classification results of k-NN for Case 1 (imbalanced: ten bad and 32 good lots; balanced: 32 bad and good lots each) and Case 2 (imbalanced: seven bad and 68 good lots; balanced: 68 bad and good lots each) in terms of ten-fold cross validation.

imbalanced instances "without augmentation" and the balanced instances with $\alpha_{\text{os}} = 100\%$ for the WAT (Case 1) and CP (Case 2) are tested. The numbers of good and bad lots can be seen in Table III. The classification results are demonstrated in Figs. 7–10. Here, the original dataset is partitioned into ten equal-sized subsamples. Of the ten subsamples, a single subsample is retained as the validation data for testing the model, and the remaining nine subsamples are used as the training data. Then, the cross-validation process is repeated ten times (i.e., folds), with each of the ten subsamples used exactly once as the validation data.

For k-NN, different numbers of the nearest neighbors (k) are tried for classification in Fig. 7 although the value of $k = 2$ should be a legitimate choice for both scenarios. In the imbalanced instance of WAT, the accuracy peaks at $k = 6$ and tapers off after $k = 9$. In the balanced instance of WAT, the accuracy reaches a maximum at $k = 2$ and fluctuates drastically afterward, where the overfitting apparently occurs. In the unbalanced instance of CP, the accuracy attains its highest value at $k = 3$ and stabilizes thereafter. In the balanced instance of CP, the accuracy reaches its highest value at $k = 2$ as expected but performs worse than the imbalanced instance. The unstable k-NN results arise from the dimensionality curse that in high dimensional spaces the distances between nearest and farthest points from designated points are indistinguishable [28].

For XGBoost, different numbers of trees are investigated as the learning rate is set at 0.3, the maximum depth is set

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

TABLE III
COMPARISON OF SMOTE RATIOS IN WAT DATA AND CP DATA

Dataset	SMOTE (α_{OS})	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
WAT data	positive data	10	-	-	11	12	16	19	22	25	28	32
	negative data	32	-	-	32	32	32	32	32	32	32	32
CP data	positive data	7	-	13	19	27	34	40	47	54	61	68
	negative data	68	-	68	68	68	68	68	68	68	68	68

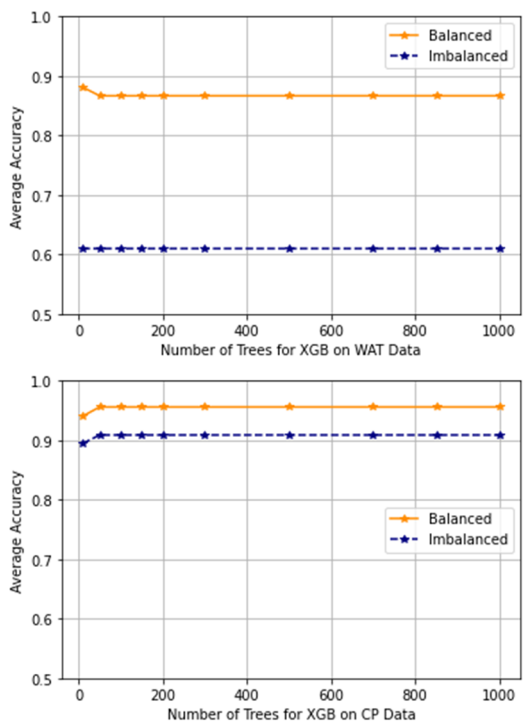


Fig. 8. Classification results of XGBoost for Case 1 (imbalanced: ten bad and 32 good lots; balanced: 32 bad and good lots each) and Case 2 (imbalanced: seven bad and 68 good lots; balanced: 68 bad and good lots each) in terms of ten-fold cross validation.

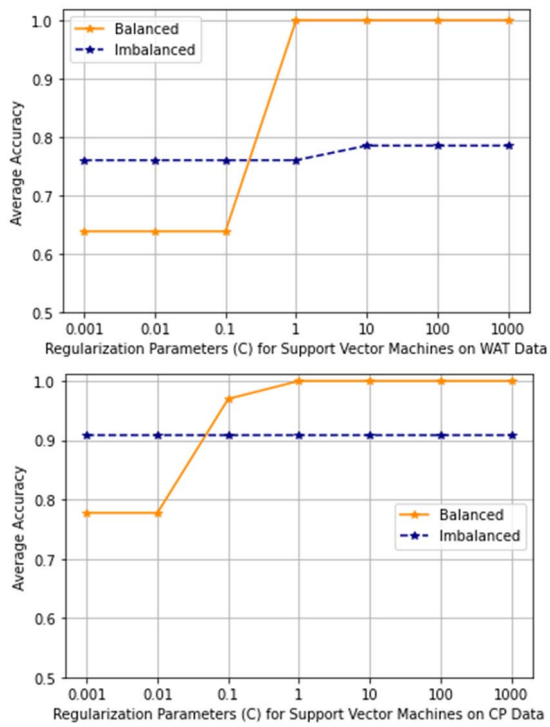


Fig. 9. Classification results of SVM for Case 1 (imbalanced: ten bad and 32 good lots; balanced: 32 bad and good lots each) and Case 2 (imbalanced: seven bad and 68 good lots; balanced: 68 bad and good lots each) in terms of ten-fold cross validation.

to six and the remaining parameters are set to default in the toolkit [27]. As can clearly be seen from Fig. 8, the data augmentation by using the SMOTE in XGBoost significantly enhances the accuracies from the unbalanced instance to the balanced one for both WAT and CP scenarios. As can be seen in the figure, XGBoost does not suffer from overfitting as the number of trees increases and a low learning rate is applied [29]. In this application, XGBoost can build on more trees without compromising the accuracy of the model. The number of ten trees is a robust choice for both scenarios for later experimentation.

For SVM, different values of the regularization parameter C are tested, serving as a degree of importance given to misclassification. If a larger value of C is given, the SVM model allows less misclassified samples by using a tighter-margin hyperplane. On the other hand, if a smaller value of C is used, the SVM model allows more misclassified samples by using a wider-margin hyperplane. The classification results of both scenarios for the unbalanced and balanced instances are shown in Fig. 9. For both scenarios, the accuracy of the SVM model greatly improves close to 1.0 as the regularization parameter is set greater than or equal to 1.0.

For LR, different values of the regularization parameter (c parameter) are examined, which serves as a penalty term against complexity added to the cost function. The c parameter is the inverse of regularization strength in LR, which is used to reduce overfitting. If a large value of c is used, the complexity of the learned model increases to adjust itself better to the data. If a smaller value of c is used, the regularization strength increases to impose a less complex model that tends to underfit

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

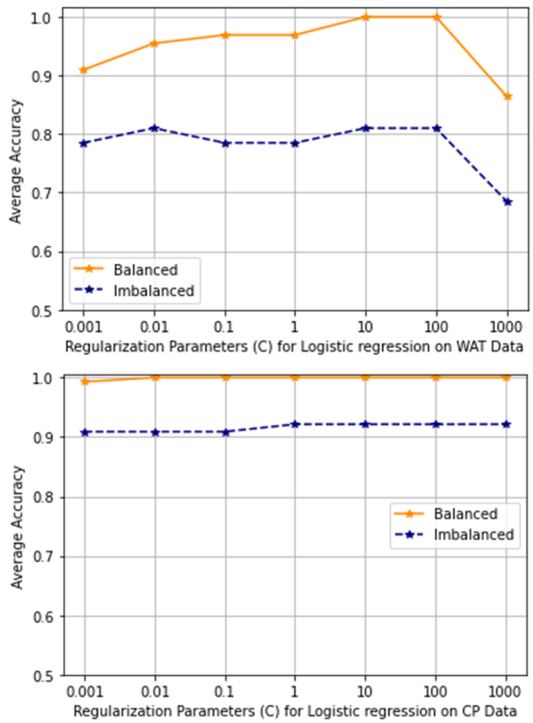


Fig. 10. Classification results of LR for Case 1 (imbalanced: ten bad and 32 good lots; balanced: 32 bad and good lots each) and Case 2 (imbalanced: seven bad and 68 good lots; balanced: 68 bad and good lots each) in terms of ten-fold cross validation.

the data. The classification results are shown in Fig. 10. Like the SVM results, the LR model improves noticeably on the classification accuracies by incorporating the SMOTE augmentation from the unbalanced to balanced case in both scenarios. As can be seen from the figure, the best parameter setting of c is 0.1. It is evident from Figs. 7–10 that the SMOTE augmentation is highly beneficial to the task of classification. The best-practice ratio for each classifier will be sought shortly.

In this section, a comprehensive comparison of the four studied classifiers for the WAT and CP scenarios, assessed by the average accuracy and F_β score ($\beta = 0.5, 1, 2$) in (8) and (9), over ten-fold cross validation is tabulated in Table IV. The SMOTE ratio ranges from 0% (without augmentation) to 100% (full augmentation) under evaluation. The number in the table indicates the average of the performance measure with standard deviation in parenthesis. The $F_{0.5}$ score places more attention on minimizing FP than minimizing FN. The F_2 score puts more attention on minimizing FN than minimizing FP. For the imbalanced datasets investigated in this article, the F_2 score appears to be a more suitable F measure than the $F_{0.5}$ and F_1 scores. In a preliminary experiment, four different kernel functions for SVM (i.e., linear, radial basis function, polynomial and sigmoid) were examined on the classification performance for the WAT and CP scenarios. Based on an overall evaluation, the linear and sigmoid kernel functions outperform the radial basis function and the polynomial kernel function. The linear kernel function is adequate for really high-dimensional data, so the linear kernel function is selected for SVM in our article. The computational SVM results corresponding to the linear kernel function are used in Table IV. The preliminary computational SVM results are available upon request.

As shown in Table IV, none of the classifiers under investigation provide satisfactory classification results in the unaugmented instances, particularly for the WAT scenario in accuracy and the CP scenario in F score. The SMOTE contributes the sharpest rise in both measures to the LR model, followed by the SVM model. The LR model forms a firm statistical foundation and a probabilistic model useful for explaining the data. The trained LR model can interpret model coefficients as indicators of feature importance.

For the WAT scenario, SVM with $\alpha_{\text{os}} = 90\%-100\%$ and LR with $\alpha_{\text{os}} = 70\%-90\%$ perform perfectly in both accuracy and F_1 score between the classification models. For the CP scenario, SVM with $\alpha_{\text{os}} = 40\%-100\%$ and LR with $\alpha_{\text{os}} = 30\%, 50\%-100\%$ perform perfectly in both accuracy and F_1 score between the classification models. All four models benefit from the data augmentation, except that the SMOTE augmentation does not contribute to the k-NN model in accuracy for the CP scenario.

Attention is now turned to the F_2 score results, where recall has a two times higher weight than precision. The F_2 score is 1.0 in the case of perfect precision and recall, and 0.0 in the case of either zero precision or zero recall. None of the four classifiers perform well in the unaugmented instance. In the CP scenario without data augmentation, the XGBoost model cannot identify any of the bad lots, causing the recall and F score to be zero in spite of a seemingly high accuracy of 0.91. Using the data augmentation of $\alpha_{\text{os}} = 30\%$, the F_2 score increases to 0.82. Based on the F_2 score results, the LR and SVM models perform comparatively well for both scenarios; the k-NN model performs worst partly due to the curse of dimensionality. It can be seen from Table IV that the LR-SMOTE takes the lead, followed by the SVM-SMOTE, and both methods perform equally well for the WAT/CP scenarios when data augmentation is used. Due to the interpretability of feature importance in LR, the LR-SMOTE is preferable to the SVM-SMOTE. In summary, the LR model augmented by the SMOTE appears to be an appropriate option for differentiating the good and bad lots of wafers in the WAT and CP scenarios investigated in this article. Based on an overall evaluation from Table IV, the LR model is selected for fault diagnosis. The results of precision and recall corresponding to Table IV are available upon request.

From Table IV, the SMOTE ratio ranging between 70% and 90% presents a competitive ability in accuracy and F score. To further verify the effectiveness of the LR model augmented by the SMOTE in differentiating the good and bad lots against the one without data augmentation, the ratio of 90% is opted over 70% for the purpose of better data visualization. Two-dimensional data visualization results of the WAT and CP scenarios are plotted in Figs. 11 and 12, respectively. In the PCA experiment, the x-axis is the first principal component and the y-axis is the second principal component after data reduction. As can be seen from Figs. 11 and 12, the SMOTE does help the LR model to separate the bad lots from the good

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

TABLE IV

COMPARISON RESULTS OF DIFFERENT CLASSIFICATION MODELS BASED ON A VARIETY OF SMOTE RATIOS

		SMOTE (α_{os})	0%	30%	40%	50%	60%	70%	80%	90%	100%
Accuracy	WAT scenario	KNN	0.73 (0.325)	0.70 (0.22)	0.78 (0.24)	0.84 (0.125)	0.82 (0.19)	0.82 (0.105)	0.52 (0.065)	0.78 (0.13)	0.61 (0.18)
		XGB	0.61 (0.235)	0.61 (0.235)	0.73 (0.145)	0.72 (0.195)	0.78 (0.165)	0.78 (0.21)	0.91 (0.06)	0.75 (0.165)	0.85 (0.155)
		SVM	0.81 (0.22)	0.70 (0.46)	0.96 (0.09)	0.98 (0.06)	0.98 (0.06)	0.97 (0.10)	0.98 (0.05)	1.00 (0.00)	1.00 (0.00)
		LR	0.81 (0.215)	0.93 (0.115)	0.96 (0.09)	0.98 (0.06)	0.98 (0.06)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.045)
	CP scenario	KNN	0.92 (0.065)	0.86 (0.13)	0.63 (0.095)	0.59 (0.08)	0.66 (0.11)	0.54 (0.08)	0.52 (0.065)	0.57 (0.11)	0.57 (0.07)
		XGB	0.91 (0.06)	0.96 (0.075)	0.96 (0.065)	0.99 (0.03)	0.97 (0.06)	0.96 (0.06)	0.97 (0.04)	0.98 (0.05)	0.97 (0.06)
		SVM	0.92 (0.06)	0.98 (0.05)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
		LR	0.92 (0.065)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
$F_{0.5}$-score	WAT scenario	KNN	0.51 (0.45)	0.47 (0.395)	0.48 (0.425)	0.58 (0.235)	0.49 (0.115)	0.66 (0.11)	0.53 (0.095)	0.57 (0.06)	0.59 (0.085)
		XGB	0.04 (0.12)	0.28 (0.435)	0.34 (0.445)	0.71 (0.37)	0.72 (0.325)	0.77 (0.315)	0.87 (0.145)	0.88 (0.18)	0.88 (0.13)
		SVM	0.48 (0.45)	0.70 (0.46)	0.80 (0.40)	0.98 (0.05)	0.97 (0.09)	0.95 (0.15)	0.95 (0.06)	1.00 (0.00)	1.00 (0.00)
		LR	0.44 (0.47)	0.60 (0.445)	0.70 (0.46)	0.98 (0.05)	0.94 (0.135)	0.97 (0.085)	0.97 (0.085)	0.97 (0.085)	0.98 (0.065)
	CP scenario	KNN	0.10 (0.30)	0.41 (0.125)	0.36 (0.06)	0.42 (0.05)	0.48 (0.05)	0.48 (0.045)	0.51 (0.04)	0.54 (0.025)	0.57 (0.015)
		XGB	0.00 (0.00)	0.93 (0.08)	0.95 (0.085)	0.98 (0.035)	0.99 (0.025)	0.97 (0.075)	0.99 (0.02)	0.99 (0.015)	0.98 (0.03)
		SVM	0.10 (0.30)	0.98 (0.05)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
		LR	0.10 (0.30)	0.98 (0.05)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
F_1-score	WAT scenario	KNN	0.51 (0.45)	0.51 (0.39)	0.51 (0.435)	0.55 (0.31)	0.60 (0.12)	0.75 (0.095)	0.62 (0.08)	0.69 (0.055)	0.69 (0.075)
		XGB	0.04 (0.12)	0.17 (0.34)	0.24 (0.40)	0.61 (0.33)	0.73 (0.32)	0.68 (0.185)	0.78 (0.30)	0.81 (0.22)	0.85 (0.115)
		SVM	0.51 (0.45)	0.70 (0.46)	0.97 (0.10)	0.98 (0.06)	0.97 (0.10)	0.98 (0.05)	0.98 (0.05)	1.00 (0.00)	1.00 (0.00)
		LR	0.44 (0.47)	0.70 (0.46)	0.80 (0.40)	0.97 (0.10)	0.97 (0.10)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.99 (0.035)
	CP scenario	KNN	0.10 (0.30)	0.54 (0.105)	0.51 (0.07)	0.56 (0.055)	0.62 (0.07)	0.62 (0.04)	0.64 (0.045)	0.68 (0.045)	0.70 (0.04)
		XGB	0.00 (0.00)	0.83 (0.165)	0.93 (0.135)	0.95 (0.085)	0.96 (0.08)	0.94 (0.085)	0.96 (0.065)	0.97 (0.08)	0.96 (0.065)
		SVM	0.10 (0.30)	0.70 (0.46)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
		LR	0.10 (0.30)	1.00 (0.00)	0.93 (0.135)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
F_2-score	WAT scenario	KNN	0.51 (0.45)	0.57 (0.405)	0.56 (0.46)	0.78 (0.265)	0.76 (0.12)	0.88 (0.055)	0.81 (0.06)	0.84 (0.035)	0.85 (0.04)
		XGB	0.04 (0.12)	0.26 (0.405)	0.35 (0.45)	0.69 (0.385)	0.74 (0.33)	0.72 (0.315)	0.83 (0.175)	0.84 (0.25)	0.89 (0.12)
		SVM	0.55 (0.46)	0.70 (0.46)	0.96 (0.09)	0.98 (0.06)	0.98 (0.06)	0.97 (0.10)	0.98 (0.05)	1.00 (0.00)	1.00 (0.00)
		LR	0.44 (0.47)	0.60 (0.49)	0.70 (0.46)	1.00 (0.00)	0.94 (0.135)	1.00 (0.00)	0.99 (0.025)	0.99 (0.025)	0.99 (0.02)
	CP scenario	KNN	0.10 (0.30)	0.72 (0.085)	0.69 (0.055)	0.74 (0.04)	0.79 (0.035)	0.79 (0.03)	0.88 (0.025)	0.83 (0.015)	0.84 (0.01)
		XGB	0.00 (0.00)	0.82 (0.22)	0.90 (0.175)	0.94 (0.115)	0.96 (0.085)	0.95 (0.095)	0.95 (0.075)	0.97 (0.055)	0.97 (0.05)
		SVM	0.10 (0.30)	0.96 (0.13)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)
		LR	0.10 (0.30)	0.96 (0.135)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)

lots in the augmented instance more than the one without augmentation. The good-lot samples roughly fall within an ellipse, and the majority of augmented bad-lot samples distribute outside the ellipse in the 2-D PCA plot. In Fig. 11(b), the first principle component accommodates 12.7% and the second principle component accommodates 6.5% for the WAT data. In Fig. 12(b), the first principle component contributes 13.6% and the second principle component accommodates 7.4% for

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

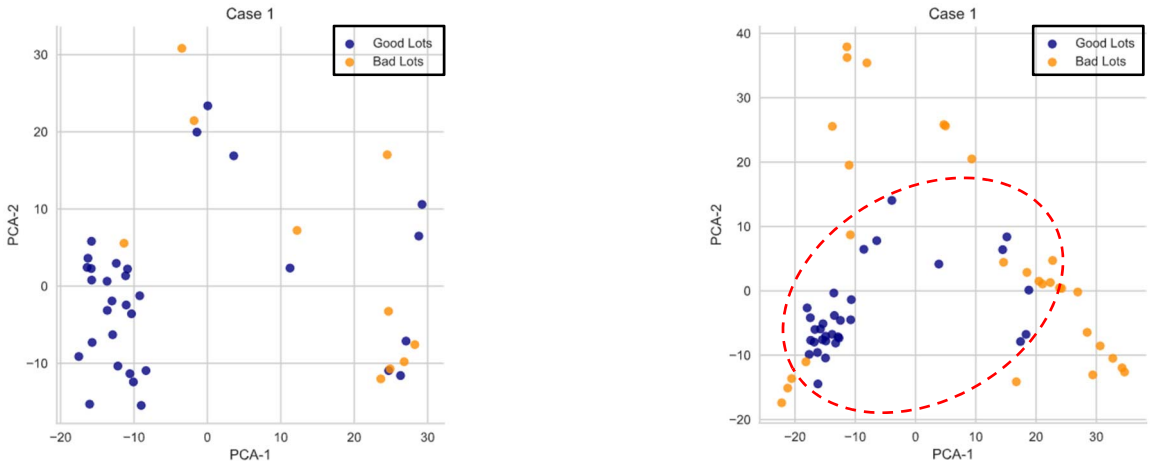


Fig. 11. PCA-2D data visualization for the WAT data: (a) without augmentation and (b) with augmentation of $\alpha_{os} = 90\%$.

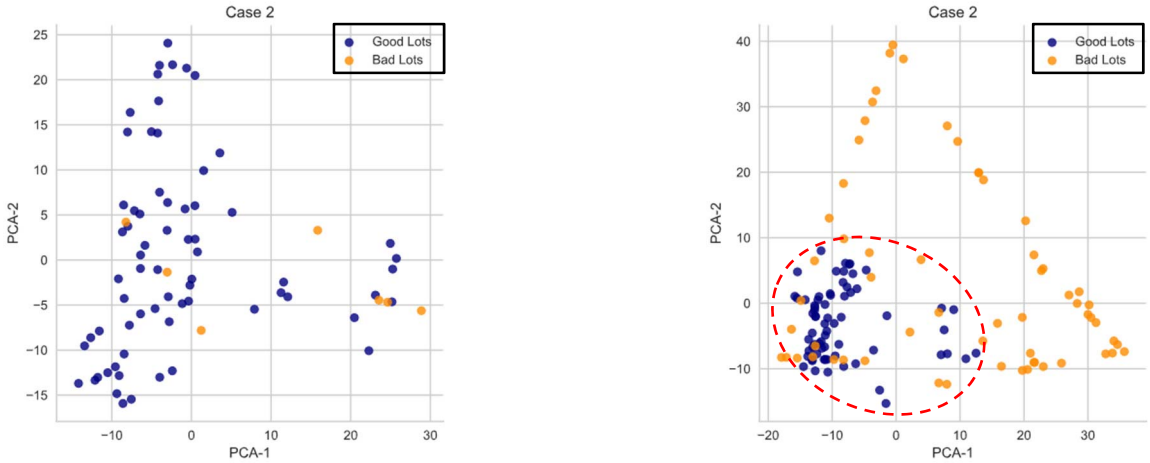


Fig. 12. PCA-2D data visualization for the CP data: (a) without augmentation and (b) with augmentation of $\alpha_{os} = 90\%$.

the CP data. Careful observation of these two figures suggests triangle patterns after applying SMOTE to the minority group. However, it has been confirmed from several preliminary experiments that no profound implication exists for the unexpected triangle patterns of the augmented minority group. With such a further verification via the data visualization by using the dimensionality reduction method, we feel confident that the proposed LR-SMOTE model can provide insightful information gained from the classification results, helping facilitate fault diagnosis in the WAT and CP scenarios, as will be shown shortly.

According to the experimental results previously presented, the capability of classification models has been greatly augmented by the SMOTE. For the primary research into the WAT data and CP data, the most essential task is to build an appropriate classification model, through which to anchor a set of "source" process parameters that separate most good and bad lots. Accordingly, the source operation, generally attributed to the faults in the WAT and CP data, can be successfully pinpointed. To that end, with the assistance of the SMOTE, the LR model wins out in classification and is chosen to identify the "root cause" of faults that have rarely occurred in the BEOL stage of semiconductor manufacturing.

By means of ten-fold cross-validation in feature importance, the top ten feature importance are selected from the LR model without using SMOTE, as hierarchically shown in Fig. 13(a) for the WAT data and Fig. 14(a) for the CP data, respectively. In these two sub-figures, the horizontal axis denotes the process parameter; the vertical axis denotes the importance. Next, the importance of every process parameter accumulates into its corresponding operation, as descendingly shown in Figs. 13(b) and 14(b), respectively. The LR model without using SMOTE identifies the operation named "1T.WWW01" that takes the lead in accumulated importance for the WAT scenario.

However, there is no significant difference in accumulated importance among the first five ranked operations. Furthermore, the failure operation named "3P.PP81A," confirmed by the domain engineers, is not in the list generated by using the LR model without using SMOTE. In Fig. 14(b), the operation named "1M.CVMW1" is ranked first in accumulated importance for the CP scenario identified by the LR model without

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

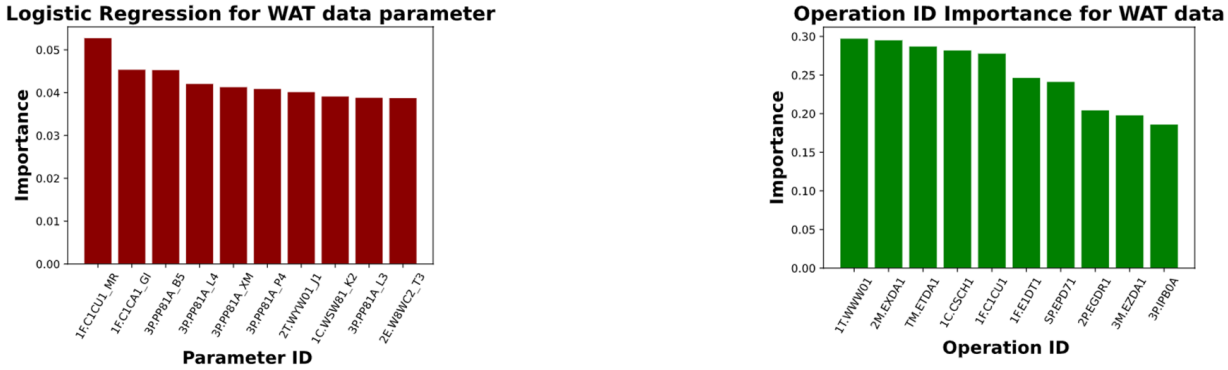


Fig. 13. LR model without using SMOTE. (a) Top-ten parameter importance in WAT data. (b) Clustering all parameters into operation from WAT data.

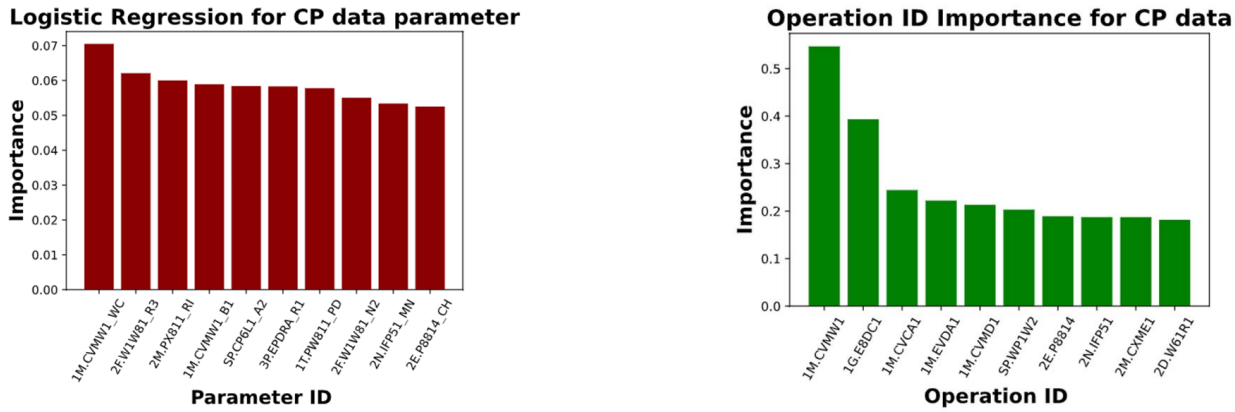


Fig. 14. LR model without using SMOTE. (a) Top-ten parameter importance in CP data. (b) Clustering all parameters into operation from CP data.

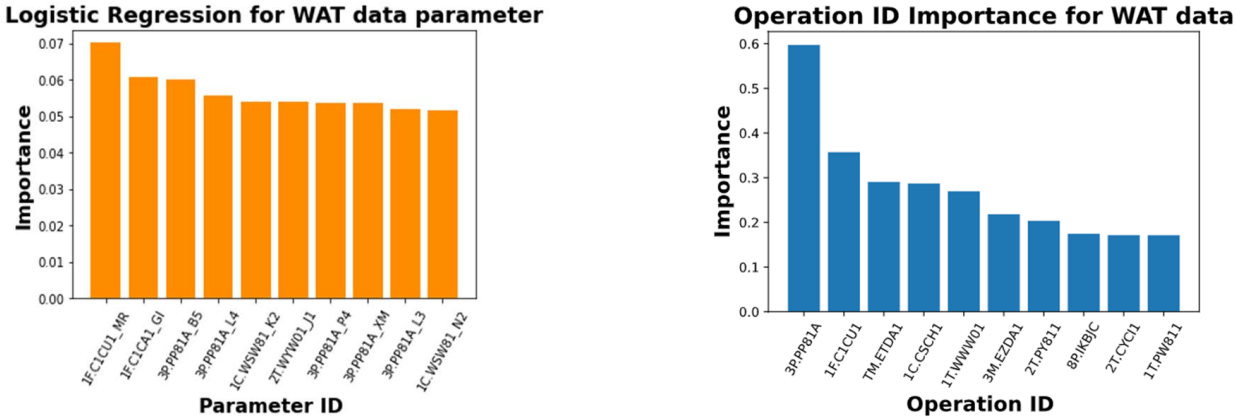


Fig. 15. LR model using SMOTE. (a) Top-ten parameter importance in WAT data. (b) Clustering all parameters into operation from WAT data.

using SMOTE, which is confirmed by the domain engineers as the "failure operation."

Subsequently, the top feature importance is screened from the LR model using SMOTE, as hierarchically shown in Fig. 15(a) for the WAT data and Fig. 16(a) for the CP data, respectively. As before, the importance of every process parameter accumulates into its corresponding operation, as shown in Figs. 15(b) and 16(b), respectively. In these two sub-figures, the operation named "3P.PP81A" takes the lead in accumulated importance for the WAT scenario; the operation named "1M.CVMW1" is ranked first in accumulated importance for the CP scenario. These two operations identified by using the proposed LR-SMOTE solution procedure are given more accumulated importance than the remaining operations. These two identified key operations have also been confirmed the failure operations by the domain engineers, corresponding to the WAT and CP scenarios. In some practical circumstances, collinearity could occur when new features are

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

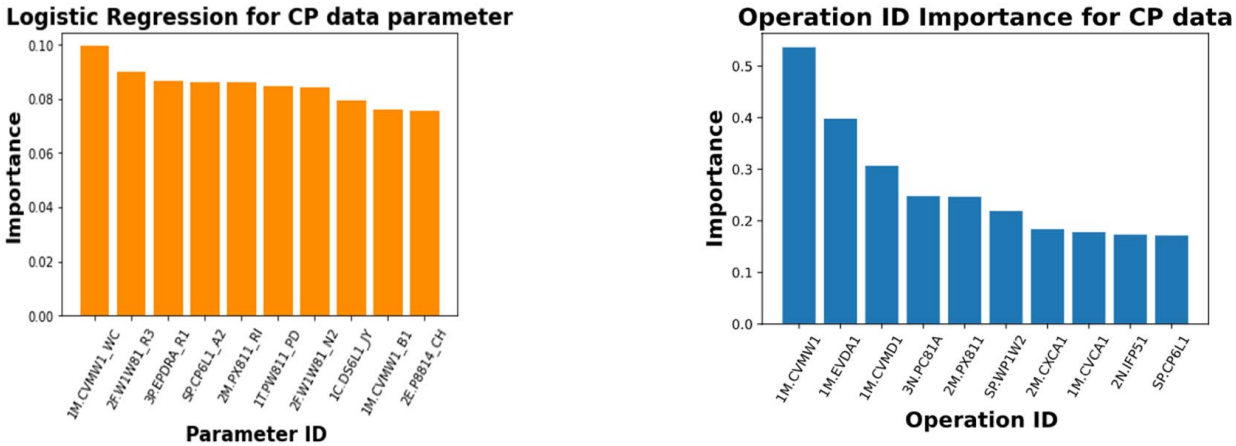


Fig. 16. LR model using SMOTE. (a) Top-ten parameter importance in CP data. (b) Clustering all parameters into operation from CP data.

TABLE V

IDENTIFIED FAILURE OPERATIONS BY MEANS OF ALL THE FEOL FEATURES

Model\Rank	1	2	3	4	5	6	7	8	9	10
XGBoost/WAT	1F.E1DN1	1C.PS811	8P.IKBJC	**3P.PP81A**	1F.C1CU1	1C.CSCH1	2I.I3P02	1G.P8811	1T.PW811	8P.EKDRC
XGBoost/CP	2E.P8814	1M.CVCA1	1T.PWB11	2I.P3811	**1M.CVMW1**	1F.C1CA1	1G.P8811	SP.CP6L2	2M.PX811	1M.CVMD1

created which are dependent on other features. In the FEOL process, redundant features may be mechanically devised in an intended manner. If found significant with comparatively high importance for the WAT or CP data, associated dependent features should reside in the same operation. Therefore, the failure operation can still be successfully pinpointed by the proposed solution procedure without undue difficulty.

Before closing this section, we bring to light an interesting comparison where the fault diagnosis is conducted by including all the features, i.e., containing the process station, test station, and measurement station data in the FEOL process. For validation purposes, additional experiments are performed without data augmentation where 42 lots of wafers having 12 915 features for the WAT data and 72 lots of wafers having 12 776 features for the CP data are used to perform the fault diagnosis, respectively. Among the compared classification models, only XGBoost identifies the true failure operation within the top ten rankings. According to the top ten feature importance, the top ten failure operations identified by XGBoost are ranked in Table V. From the fault diagnosis results shown in the table, the "true" failure operation (**3P.PP81A**) in the WAT data is only ranked fourth, the "true" failure operation (**1M.CVMW1**) in the CP data is only ranked fifth. This additional experimental result stresses the merit of the data augmentation strategy used in the LR-SMOTE model, proving to be a viable tool for resolving the challenge of failure diagnosis between the FEOL and BEOL processes.

In summary, the proposed framework presents a scientific, data-driven solution procedure for fault diagnosis for the WAT and CP data in the semiconductor manufacturing process without resort to a detailed examination of tedious and time-consuming internal process quality control (IPQC) records. The ML techniques proposed for the use of classification and data visualization in this article have shown their effectiveness for fault diagnosis under extremely high feature dimensionality with computational efficiency and easy implementation. Through the real-data empirical study conducted here, the proposed fault diagnosis framework proves to be a viable solution procedure for fault prognosis in semiconductor manufacturing.

V. CONCLUSION

Semiconductor manufacturing is the core and foundation of the electronic components industry, categorized into two main lines. The FEOL is typically called the foundry, where the individual devices are patterned and fabricated; the BEOL is where interconnections are created with wiring on the wafer, including bonding and packaging. This article focuses on the WAT and CP processes where testing parameters are collected and used for identifying wafer defects during the BEOL. The objective of this article is to perform a fault diagnosis of the WAT and CP data by tracing upstream process parameters and corresponding operations in the FEOL.

Without loss of generality, the WAT and CP scenarios have been addressed simultaneously by proposing a new fault diagnosis framework where a series of ML techniques for classification and data visualization are used in the presented solution procedure. By a comprehensive comparison of the four selected classifiers, the proposed LR-SMOTE model is chosen for the classification assignment in combination with vector-valued data augmentation and PCA. In the empirical study, the WAT and CP data provided by a world-class semiconductor company in Taiwan are used to illustrate the effectiveness of fault diagnosis delivered by the proposed solution procedure. The identified key parameters and key operations that produce faults in the WAT and CP scenarios were validated by the domain engineers. On the above grounds, we conclude that the new ML solution procedure

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

shows great promise as a viable tool for fault diagnosis of WAT and CP in semiconductor manufacturing.

The primary contribution of this article is the proposal of an integrated solution framework, combining ML models, data augmentation, and data visualization, to bridge the gap between the FEOL and BEOL processes. A comprehensive trial-and-error procedure presented in this article is recommended for model selection without prior knowledge. In this study, k-NN fails in modeling due to the curse of dimensionality and should be excluded from consideration. On the other hand, the comprehensibility of modeling results aids the success of practical implementation. In addition to domain knowledge, the interpretability of feature importance helps to gain understanding of how the predictive capability of the learning algorithm can be enhanced and the APC should be deployed. A smaller set of key features selected from an abundant range of features should capture accurate information that is not apparent in the original set. While analyzing a brand new WAT/CP dataset, the proposed fault diagnosis procedure provides the process engineer with a top-ten list of potential failure operations. Through the list, the process engineer can examine how crucially every identified operation has an impact on the bad lots by the order of hierarchical importance. If one of those operations is identified as "the true failure operation" but not listed at the very top, then the field test results should be fed back to the data analyst for model tuning/re-training by conducting further parameterization or data collection/augmentation.

Building upon this research, there are several interesting topics for future scrutiny in this area. For instance, it will be interesting to investigate alternative means for accounting for the interrelationship between the status of the process parameters [also known as status variable identification (SVID)] in the FEOL and the test yield in the BEOL. If successfully done, yield prediction would be practicably feasible before the WAT and CP processes are executed. Another potential research work could be how to develop an appropriate version of the generative adversarial network (GAN) [30], [31] to augment the time series data of the process stations corresponding to the faulty wafers. Alternatively, synthetically generating the time series data of faulty wafers by using the variational autoencoder [32] also shows promise in data augmentation.

REFERENCES

[1] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, May 2017.

[2] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006, .

[4] D. Freedman, *Statistical Models: Theory and Practice*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[5] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, 2017.

[6] X. Gu, F.-L. Chung, H. Ishibuchi, and S. Wang, "Imbalanced TSK fuzzy classifier by cross-class Bayesian fuzzy clustering and imbalance learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2005–2020, Aug. 2017.

[7] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017.

[8] B. M. Haddad, S. Yang, L. J. Karam, J. Ye, N. S. Patel, and M. W. Braun, "Multifeature, sparse-based approach for defects detection and classification in semiconductor units," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 145–159, Jan. 2018.

[9] M. Bader-El-Den, E. Teitei, and T. Perry, "Biased random forest for dealing with the class imbalance problem," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2163–2172, Jul. 2019.

[10] C.-M. Fan, R.-S. Guo, S.-C. Chang, and C.-S. Wei, "SHEWMA: An end-of-line SPC scheme using wafer acceptance test data," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 3, pp. 344–358, Aug. 2000.

[11] H.-C. Hong and L.-Y. Lin, "Accurate and fast on-wafer test circuitry for device array characterization in wafer acceptance test," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3467–3479, Sep. 2019.

[12] H. Xu, J. Zhang, Y. Lv, and P. Zheng, "Hybrid feature selection for wafer acceptance test parameters in semiconductor manufacturing," *IEEE Access*, vol. 8, pp. 17320–17330, 2020.

[13] Y.-C. Chang, P.-Y. Wang, D.-C. Chang, and S. S. H. Hsu, "Wide-band conducted electromagnetic emission measurements using IPD chip probes," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 11, pp. 4063–4070, Nov. 2016.

[14] Q. Liu, A. C. F. Reniers, U. Johannsen, M. C. van Beurden, and A. B. Smolders, "Improved probing reliability in antenna-on-chip measurements," *IEEE Antennas Wireless Propag. Lett.*, vol. 17, no. 9, pp. 1745–1749, Sep. 2018.

[15] T. Lee, K. B. Lee, and C. O. Kim, "Performance of machine learning algorithms for class-imbalanced process fault detection problems," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 4, pp. 436–445, Nov. 2016.

[16] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 315–322, May 2018.

[17] E. Kim, S. Cho, B. Lee, and M. Cho, "Fault detection and diagnosis using self-attentive convolutional neural networks for variable-length sensor data in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 302–309, Aug. 2019.

[18] S.-K.-S. Fan, C.-Y. Hsu, D.-M. Tsai, F. He, and C.-C. Cheng, "Data-driven approach for fault detection and diagnostic in semiconductor manufacturing," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1925–1936, Oct. 2020.

[19] G. May and S. Sze, *Fundamentals of Semiconductor Fabrication*. Hoboken, NJ, USA: Wiley, 2003.

[20] J. N. D. Gupta, R. Ruiz, J. W. Fowler, and S. J. Mason, "Operational planning and control of semiconductor wafer production," *Prod. Planning Control*, vol. 17, no. 7, pp. 639–647, Oct. 2006.

[21] D.-H. Lee, J.-K. Yang, C.-H. Lee, and K.-J. Kim, "A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data," *J. Manuf. Syst.*, vol. 52, pp. 146–156, Jul. 2019.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[23] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

[24] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441 and 498-520, 1933.

[25] K. Song, F. Nie, J. Han, and X. Li, "Rank-k 2-D multinomial logistic regression for matrix data classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3524–3537, Aug. 2018.

[26] Z.-Y. Yang, Y. Liang, H. Zhang, H. Chai, B. Zhang, and C. Peng, "Robust sparse logistic regression with the L_q $(0 < q < 1)$ regularization for feature selection using gene expression data," *IEEE Access*, vol. 6, pp. 68586–68595, 2018.

[27] Python Software Foundation. *Python Language Reference, Version 3.6*. Accessed: Oct. 17, 2020. [[Online]. Available: http://www.python.org

[28] N. Kouiroukidis and G. Evangelidis, "The effects of dimensionality curse in high dimensional kNN search," in *Proc. 15th Panhellenic Conf. Informat.*, Kastoria, Greece, Sep. 2011, pp. 41–45.

[29] L. Zhang and C. Zhan, "Machine learning in rock facies classification: An application of XGBoost," in *Proc. Int. Geophys. Conf.*, Qingdao, China, Apr. 2017, pp. 1371–1374.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

[30] K. Yan, J. Su, J. Huang, and Y. Mo, "Chiller fault diagnosis based on VAE-enabled generative adversarial networks," *IEEE Trans. Autom. Sci. Eng.*, early access, Nov. 19, 2020, doi: 10.1109/TASE.2020.3035620.

[31] D.-M. Tsai, S.-K.-S. Fan, and Y.-H. Chou, "Auto-annotated deep segmentation for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021, doi: 10.1109/TIM.2021.3087826.

[32] S.-K.-S. Fan, C.-Y. Hsu, C.-H. Jen, K.-L. Chen, and L.-T. Juan, "Defective wafer detection using a denoising autoencoder for semiconductor manufacturing processes," *Adv. Eng. Informat.*, vol. 46, Oct. 2020, Art. no. 101166.

Shu-Kai S. Fan received the Ph.D. degree in industrial, manufacturing, and systems engineering from the University of Texas at Arlington, Arlington, TX, USA, in 1996.

He is currently a Professor with the Department of Industrial Engineering and Management, National Taipei University of Technology (NTUT), Taipei, Taiwan. His research interests include quality engineering, image processing, big data analytics, machine/deep learning, and advanced process control of semiconductor manufacturing.

Dr. Fan is currently serving as the Editor-in-Chief for *Engineering Optimization* (Taylor & Francis) and has been the Convener of the Industrial Engineering and Management (IEM) Program, Ministry of Science and Technology (MOST), Taipei, since 2020.

Chun-Wei Cheng received the M.S. degree in industrial engineering and management from the National Taipei University of Technology (NTUT), Taipei, Taiwan, in 2020.

His research interests include advanced process control of semiconductor manufacturing and deep/machine learning applications.

Du-Ming Tsai received the B.S. degree in industrial engineering from Tunghai University, Taichung, Taiwan, in 1981, and the M.S. and Ph.D. degrees in industrial engineering from Iowa State University, Ames, IA, USA, in 1984 and 1987, respectively.

From 1988 to 1990, he was a Principal Engineer with Digital Equipment Corporation, Taiwan Branch, where his work focused on process and automation research and development. He is currently a Professor of Industrial Engineering and Management with Yuan-Ze University, Taoyuan, Taiwan. His research interests include machine vision, automated visual inspection, and deep-learning applications.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 09,2023 at 05:35:40 UTC from IEEE Xplore. Restrictions apply.

Wafer Defect Pattern Labeling and Recognition Using Semi-Supervised Learning

Katherine Shu-Min Li, *Senior Member, IEEE*, Xu-Hao Jiang, Leon Li-Yang Chen, Sying-Jyan Wang, *Member, IEEE*, Andrew Yi-Ann Huang, Jwu E. Chen, *Member, IEEE*, Hsing-Chung Liang, *Member, IEEE*, and Chun-Lung Hsu, *Member, IEEE*

***Abstract*—Wafer map defect patterns provide valuable information for root cause analysis and yield learning. Previous studies show that supervised machine learning (ML) methods can achieve good defect pattern recognition rate. However, the effectiveness of these methods rely on accurately labeled samples, which leads to two problems. First, oftentimes there are mislabeled data, which may prevent an accurate model from being established. Secondly, defect patterns that are not defined a priori will not be recognized. In this paper, we propose a semi-supervised learning method to deal with these problems. Labeled wafer maps are first used to train a prediction model, and questionable samples are excluded in the process. The prediction model is then used to classify unlabeled data. The remaining data that cannot be properly classified are then sent to an unsupervised learning algorithm to extract more defect patterns for enhanced labeling. The proposed approach is validated using the WM-811K database. With the proposed method, we are able to define five new defect pattern types, and the 14 defect types can be recognized with high accuracy.**

***Index Terms*—Wafer map, defect pattern, yield learning, unsupervised machine learning.**

I. Introduction

INTEGRATED circuit (IC) fabrication is a very complicated process, and all fabrication steps are subject to errors. The results of wafer test are usually illustrated by using a wafer map, which gives locations of failed dies in the wafer. Wafer defect patterns are the results of various problems in the fabrication and wafer test process. Thus, these patterns can be used to identify sources of errors in the process.

Local defect pattern recognition has attracted a lot of research interests recently. Early methods applied statistical analysis to wafer map defect patterns [1]. The emergence of machine learning (ML) technology makes it possible to apply ML techniques for wafer map defect pattern recognition. Two categories of ML algorithms have been developed: unsupervised learning and supervised learning.

Unsupervised learning approaches [2], [3], [4], [5], [6], and [7] can achieve automatic wafer defect patterns classification and recognition. However, these methods still rely on manual extraction of sample features which may affect the accuracy. With supervised learning, labeled samples are used to train ML models that are later used for defect pattern recognition [2], [8], [9], [10], [11], [12], [13], [14], [15], and [16]. In general, the supervised learning algorithms provides better performance than the unsupervised ones. Still, the achievable classification accuracy is not great, and it can be attributed to several problems in the labeling process. First, labeling is made by test engineers, and they may make incorrect labeling from time to time. The labeling process is subjective, and similar wafer maps can be assigned to two or even more defect types by the same person at different times. The second problem is that some wafer maps contain multiple defect patterns, but the labeling is usually carried out according to a given priority order. The existence of such wafers in the training data may lead to an inferior mode for misclassifications. The last reason is inadequate samples in some defect types. The recognition rates for defect patterns with limited wafer maps tend to be lower. Furthermore, some types are not labeled in the first place.

In order to deal with the aforementioned problems, a semi-supervised framework was proposed to achieve automatic classification of wafer defect patterns [17]. The process starts with a partially labeled dataset, and the labeled data are first used to train a prediction model. The prediction model is then used to classify unlabeled data. The remaining data that cannot be properly classified are then sent to an unsupervised learning algorithm to achieve enhanced labeling.

In this paper, we discuss the complete semi-supervised learning method and give a detailed analysis of the proposed method. The performance of the proposed method is evaluated by using the WM-811K database, including the labeled and

Manuscript received December 6, 2021; revised January 28, 2022; accepted March 10, 2022. Date of publication March 14, 2022; date of current version May 5, 2022. *(Corresponding author: Sying-Jyan Wang.)*

Katherine Shu-Min Li, Xu-Hao Jiang, and Leon Li-Yang Chen are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan (e-mail: smli@cse.nsysu.edu.tw; tyst89033@yahoo.com.tw; mailtoleon@gmail.com).

Sying-Jyan Wang is with the Department of Computer Science and Engineering, National Chung Hsing University, Taichung 40227, Taiwan (e-mail: sjwang@cs.nchu.edu.tw).

Andrew Yi-Ann Huang is with the Department of Wafer Test, NXP Semiconductors Taiwan Ltd., Kaohsiung 81170, Taiwan (e-mail: andrew.huang@nxp.com).

Jwu E. Chen is with the Department of Electrical Engineering, National Central University, Chungli 32001, Taiwan (e-mail: jechen@ee.ncu.edu.tw).

Hsing-Chung Liang is with the Department of Electrons Engineering, Chung Yuan Christian University, Chungli 32023, Taiwan (e-mail: hcliang@cycu.edu.tw).

Chun-Lung Hsu is with the Division for Design Automation Technology, Information and Communication Research Laboratories, Industrial Technology Research Institute, Hsinchu 31057, Taiwan (e-mail: clh@itri.org.tw).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSM.2022.3159246.

Digital Object Identifier 10.1109/TSM.2022.3159246

0894-6507 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

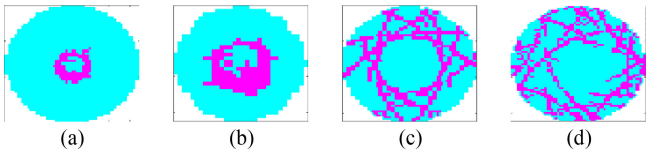


Fig. 1. Examples of inconsistent labeling in WM-811K: (a) Center, (b) Donut, (c) Donut, (d) Scratch.

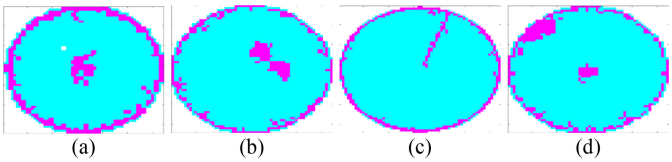


Fig. 2. Examples of mixed defect patterns in a single wafer map: (a) Center + Edge-ring, (b) Edge-ring + Local, (c) Edge-ring + Scratch, (d) Center + Edge-ring + Edge-local.

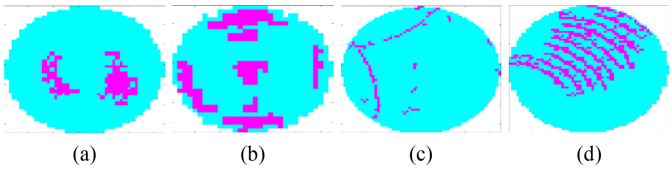


Fig. 3. Examples of multiple defect patterns of the same type: (a) two Locals, (b) multiple Locals, (c) two Scratches, (d) multiple Scratches.

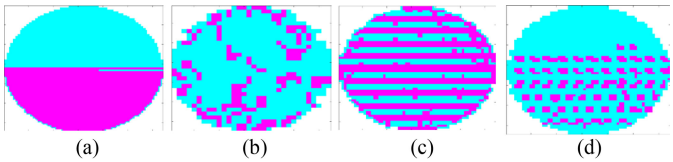


Fig. 4. Examples of special patterns.

unlabeled wafer maps in the database. Experimental results show that total 14 defect types with five new defect types can be recognized with overall accuracy of 94.37%.

II. Motivation

The performance of a prediction model is affected by the ML algorithm used as well as the quality of the training dataset. An inconsistent training set will adversely affect the performance of the prediction model regardless of the ML algorithm.

Various labeling errors may appear in a set of labeled wafer maps. For example, nine defect types have been identified for the WM-811K database: *Center*, *Donut*, *Edge-local*, *Edge-ring*, *Local*, *Near-full*, *Random*, *Scratch* and *None* [9]. Seven of the nine defect types with visually recognizable patterns can be recognized with relatively good recognition results (higher than 80%). However, a closer look at the database reveals many questionable labels, as shown in this section.

A. Inconsistent Labeling

Some examples of inconsistent labeling are given in Fig. 1. Fig. 1(a) is classified as *Center* and Fig. 1(b) is labeled as *Donut* even though they look very similar. Both Fig. 1(c) and Fig. 1(d) should be *Scratch* type, but Fig. 1(c) is classified as *Donut* instead. Obviously, Fig. 1(a) and Fig. 1(c) are mislabeled, which creates challenges for supervised learning.

B. Multiple Patterns

A more complicated wafer map may contain multiple defect patterns. The patterns may belong to the same type or they consist of mixed defect types. For example, Fig. 2 provides examples of complicated wafer maps with mixed defect pattern types, while Fig. 3 gives examples of wafer maps consisting of multiple defect patterns of the same type.

C. Special Rare Patterns

Some special defect patterns rarely appear. As a result, they are not properly identified as an individual pattern type. In most cases, such wafer maps are classified into a similar but not precise defect type. For example, the four wafer maps in Fig. 4 are labeled as *Local*, *None*, *Scratch*, and *Local*, respectively.

Actually, each of the four defect patterns in Fig. 4 can be classified as an individual defect pattern type. Each pattern appears repeatedly in WM-811K, although the total number of each pattern is relatively low. Fig. 4(a) should be classified as *Half-full*, in which almost half of the wafer are defective dies. There is no visible pattern in Fig. 4(b), but the defect density sits between *None* (low density) and *Random* (high density). The defect pattern in Fig. 4(c) consists of parallel (or vertical) lines, while in Fig. 4(d) failed dies form grid-like clusters.

D. Semi-Supervised Framework

The effectiveness of supervised learning depends on accurate labeling of the training data. However, examples in this section clearly show that labels assigned by engineers can be inappropriate in many ways. In this work, we propose to deal with the problem through semi-supervised learning. The initial manually labeled data are used to construct a prediction model through supervised learning. The model is then used to classify all the other unlabeled data. In addition, they are used to select labeled samples not fitting the model, and such samples will be processed by the unsupervised algorithm to create new defect types if they cannot be assigned to an existing type.

The enhanced labeling are then used to train the supervised learning algorithm in the next iteration. The entire process may repeat several times to achieve the final enhanced labeling and classification. The advantage of this approach is that the requirement for human efforts is relatively low, while inappropriate labeling can be corrected in the process. The overall accuracy will be improved as a result.

In order to determine whether a given label is appropriate, we employ an ensemble learning algorithm as the supervised learning method. An ensemble learning method is usually constructed from several weak learners, and the final prediction is achieved by evaluating the results obtained from the weak learners. Therefore, we can determine the likelihood of a wrong label by checking all the weak learners' results.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

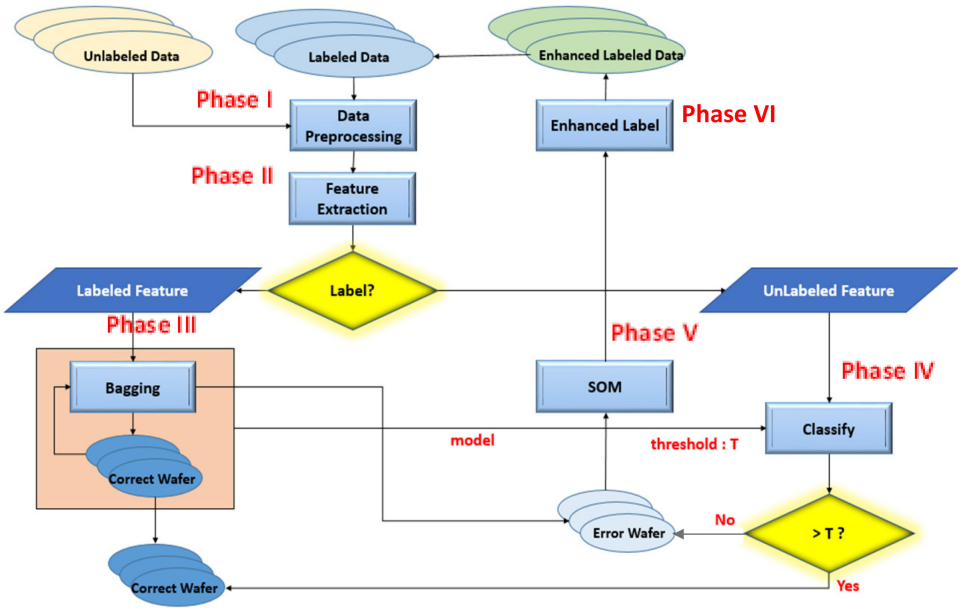


Fig. 5. Semi-supervised framework of wafer map pattern recognition with enhanced labeling.

III. Methodology

The overall flow in the proposed semi-supervised framework is illustrated in Fig. 5. The input database consists of labeled and unlabeled wafer maps. All wafers are preprocessed in Phases I and II such that useful features are extracted for the machine learning algorithms, and then the labeled wafer maps are used to build a prediction model in Phase III.

Since there are mislabeled and/or ambiguous labeling in the database, some wafer maps may not fit in the model. The prediction model constructed in Phase III is then used to classify the unlabeled wafer maps in Phase IV.

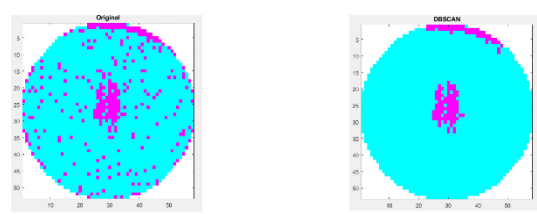
Wafers that can be classified with enough confidence (larger than threshold T) are listed as "correct wafers", while the others are "error wafers". We will then apply the unsupervised learning technique called self-organizing map (SOM) to classify the error wafers in Phase V. Additional types are constructed to label the error wafers in Phase VI. The enhanced labeled data are combined with the correct wafers to form the new labeled data. The process will be repeated through several iterations.

A. Phase I: Preprocessing

WM-811K database contains both labeled and unlabeled data, and all the data will be used in this paper. Three tasks are carried out in this phase.

1) Data Filtering: Some items in WM-811K are not specified properly. In all, roughly 1.84% of the wafer maps are discarded due to erroneous format or missing data in the wafer maps.

2) Denoising: The second task is to remove random defects in a wafer map such that a clean pattern can be extracted. An example is shown in Fig. 6(a), which contains two local defect patterns and some randomly distributed defects. The task is achieved by using the Density-based spatial clustering of applications with noise (DBSCAN) [18]. DBSCAN retrieves clusters by grouping together points in high density areas, as shown in Fig. 6(b). Points not falling in any cluster are treated as random defects and will be discarded.



(a) (b)

Fig. 6. (a) Original wafer map, (b) after denoising with DBSCAN.

3) Standardization: The wafer maps in WM-811K are given in various sizes. To facilitate machine learning, the wafers are transformed into four standard sizes: 32×32, 64×64, 128×128, and 256×256.

B. Phase II: Feature Extraction

Features that will be used by the supervised and unsupervised machine learning algorithms are extracted in this phase. Three categories of features are used.

The first category is the geometric features. First, we need to locate clusters of defects in a map, and this is achieved through connected-component analysis (CCA). For example, two connected components can be found in Fig. 6(b). The geometric features include the number of connected components and geometric information related to the two largest components in the wafer map. For the largest two components, the following features are extracted for each component: (1) distances from component center to wafer center and wafer edge, (2) area, (3) eccentricity, and (4) solidity.

The second category includes the statistic features. We employ four classes of statistic features. (1) The first class is

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

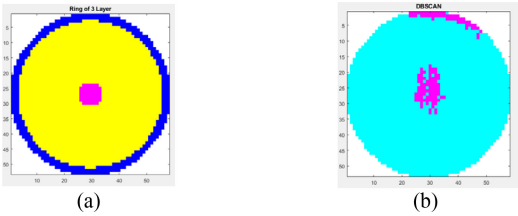


Fig. 7. (a) Three circular areas, (b) two components in edge and center.

related to the entire wafer, and the feature is the ratio of defective dies in the whole wafer. (2) The second class contains information about individual components. First we calculate the size of each component vs. all defective dies. Then, the mean, variance, maximum and minimum of the ratios are used as the features in the second class. (3) The third class is the image moment [19], also known as Hu's invariant [19]. (4) The fourth class contains defect density (DD) and defect clustering (DC) in three circular areas [20]: edge, local, and center, as shown in Fig. 7. In Fig. 7(a), the edge, local, and center areas are marked by blue, yellow, and pink, respectively. Many defect pattern types are related to the location of the major components, including *Edge-local*, *Local*, *Edge-ring*, and *Center*. Thus, these features can be used to identify such defect patterns. For example, the two components in Fig. 7(b) cover the edge and center areas.

The third category is related to Radon Transform [21] of the wafer map. The Radon Transform of a wafer map is a matrix, which can be illustrated by a figure. Fig. 8(a) gives Radon Transform of nine typical WM-811K defect patterns.

Each transformed matrix in Fig. 8(a) consists of 20 row vectors. We can use the maximum, mean, and standard deviation of the 20 row vectors as the Radon features. Each chart in Fig. 8(b) plots Radon values of the features for all row vectors in a defect pattern. In each chart, the blue, red, and green curves give the maximum, mean, and standard deviation of the 20 row vectors. It can be seen that such features are also very useful in terms of distinguishing these patterns.

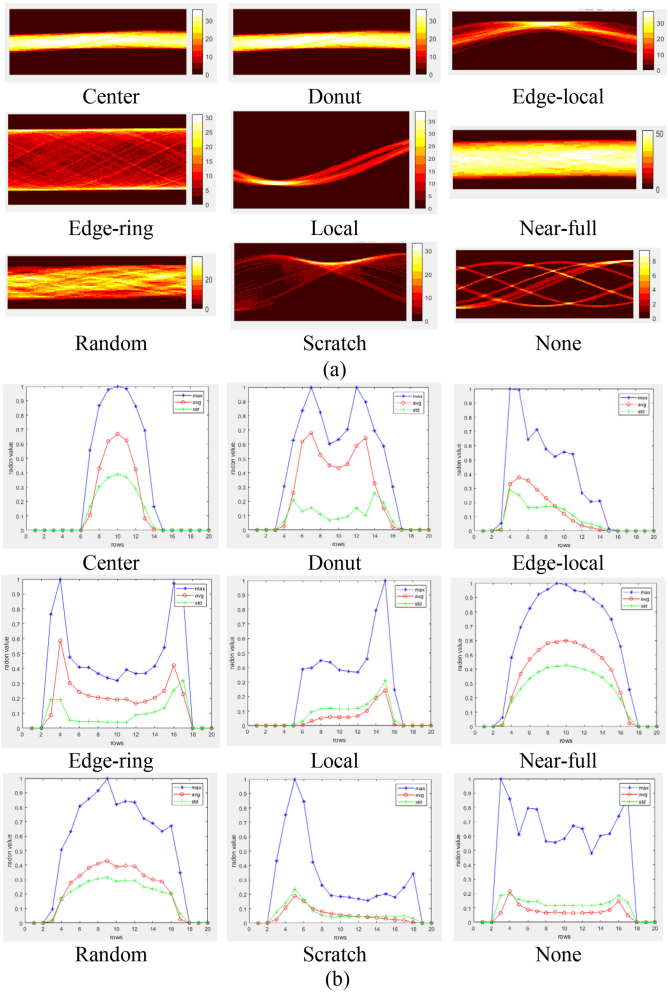


Fig. 8. (a) Radon Transformation of defect patterns, (b) maximum, minimum, and variance of the 20 row vectors of Radon transform matrices.

C. Phase III: Classification Model Construction

The goal of Phase III is to build a low-tolerance classification model according to labeled data that are currently available. Since the tolerance threshold is very low, misclassified data in WM-811K can also be screened in this step.

In order to achieve this goal, we need a classifier that can provide the confidence level of a classification result, and this goal can be achieved by employing an ensemble learning method for supervised learning. An ensemble method employs multiple learning algorithms to achieve better predictive performance. There are several ways to build an ensemble classifier [22], and in this work we use the Bootstrap aggregating, also called Bagging, for supervised learning.

In the proposed Bagging algorithm, we employ 1,000 decision trees as the sub-classifiers. The training data for each decision tree are collected by randomly selecting 80% of all training data. The final classification is determined by all decision trees through voting. A sample will be classified to defect type d if the largest number of decision trees designate it to type d. Assume that k trees classify the sample as type d, then the probability of the sample belonging to type d is $k/1000$.

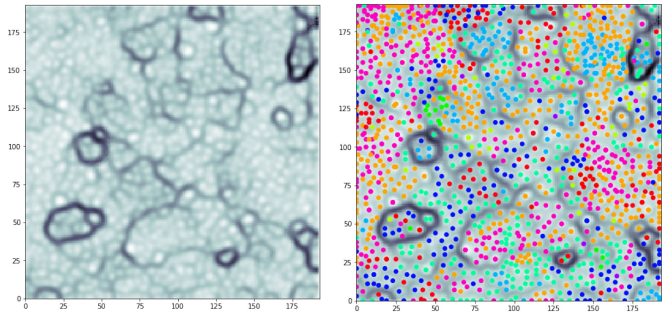
The prediction model is then used to classify all the labeled data to screen out misclassified data as error wafers. The remaining data are then applied to the Bagging algorithm again to build a new model. The final prediction model is obtained by repeating the above process 10 times.

The confidence threshold T is obtained from the final model as follows. (1) Every sample is classified by the model, and probability that the sample is assigned to the corresponding defect type is recorded. (2) For each defect type d, the mean σ_d and standard deviation σ_d of the probability values of all samples in type d are calculated. (3) The threshold of confidence level for defect type d is then set to μ_d–$3\sigma_d$, which implies that 99.9% of the samples can be correctly classified.

D. Phase IV: Classifying Unlabeled Data

The prediction model built in Phase III is used to classify unlabeled data in this phase. An unlabeled wafer map will be assigned to type d if its probability is higher than the threshold obtained in the previous phase. If a wafer map cannot be

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.



(a) (b)

Fig. 9. Unsupervised learning by SOM, (a) U-matrix, (b) U-matrix with data.

associated with any known defect type, it is classified as "error wafer".

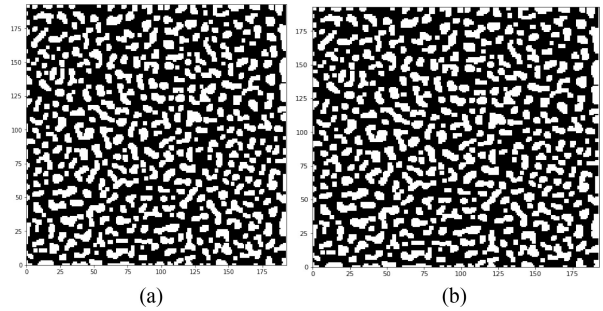
E. Phase V: Unsupervised Learning

In this phase, we will cluster similar wafer maps, which are not currently labeled into defined defect types, into similar groups. The task is carried out by self-organized map (SOM), an artificial neural network (ANN) model trained using unsupervised learning [23]. An SOM maps a set of n-dimensional input data to a two-dimensional lattice of computational units called neurons. Input data with common features are mapped to the same or neighboring neurons so as to preserve the topological properties of the input data. This makes SOMs useful for visualization of the input data. In this work, the unsupervised learning is realized using MiniSom, which is a minimalistic and Numpy based implementation of the SOM.

The procedure of SOM is outlined as follows. First, the features obtained in Phase II are represented by a characteristic matrix W of size $m \times n$, where m and n are the number of unlabeled wafers and the number of features, respectively. The matrix is normalized by multiway normalization [24]. The normalized matrix is then used to train the SOM, and the result is the U-matrix shown in Fig. 9(a).

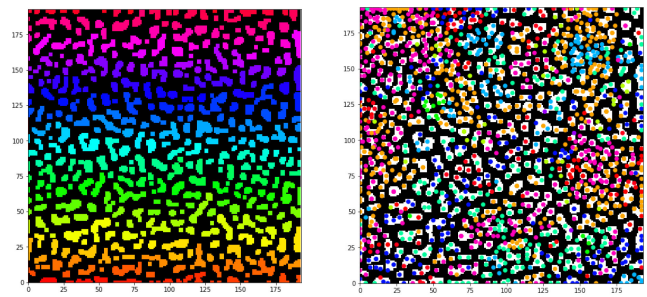
The grayscale gives the difference among weights of neighboring neurons, with darker shades meaning larger differences. In Fig. 9(b), all the wafer maps are mapped to the U-matrix, and each color represents a distinct defect pattern. It is clear that wafer maps in the same defect type tend to cluster in a neighboring area. It can be seen that the differences among neighboring clusters are not clear enough in Fig. 9(a). The grayscales are not fixed, and it is difficult to use a fixed threshold to build the walls. In order to distinguish different clusters, we propose a two-step process to build clear walls that separate white areas. The first technique is the *median wall finding*, which is used to determine the locations of walls. Fig. 10(a) gives the results of applying median wall finding to the U-matrix shown in Fig. 9(a). The second one is the *closing technique* developed in image processing. In general, closing is used to remove holes between walls to create clear boundary among white areas. The results are shown in Fig. 10(b).

At this stage, we have a black and white image corresponding to a matrix containing 0 and 1 only. Next we need to label



(a) (b)

Fig. 10. Wall building, (a) median wall finding, (b) closing.



(a) (b)

Fig. 11. (a) CCA, (b) CCA with wafer data.

individual areas in the image, and this is achieved through connected component analysis (CCA). Fig. 11(a) gives the results of CCA labeling obtained by the two-pass algorithm [25]. Each component is assigned a distinct color, and the labeling algorithm processes components from left to right, top to bottom. However, due the large number of areas in the figure, colors in horizontally neighboring areas are very close. Finally, the labeled wafer maps used in the same iteration are projected to the 2D map given in Fig. 10(b), and the result is given in Fig. 11(b). Each point in Fig. 11(b) represents a wafer map and the color gives its defect type. In most of the cases, points in the same cluster have the same color. When the color of a point is different from other points in the same cluster, the corresponding wafer map is likely misclassified in the first place.

Fig. 12 show some clusters defined by SOM. It can be seen that wafer maps in the same cluster share the same pattern.

F. Phase VI: Enhanced Labeling

The new SOM groups need to be labeled by human. Some groups are very similar to existing defect pattern types, so they will be incorporated into the existing types. Otherwise, new defect labels will created for the new defect groups, and the wafer defect database is updated accordingly.

In Fig. 13, we define five new defect pattern types in WM-811K: *Half-random*, *Center+Edge-local*, *Center+Local*, *Edge-ring+Local*, and *Normal*. The first one (*Half-random*) consists of a simple pattern, while the next three wafer maps consist of two defect patterns. Examples of these defect pattern types are given in Fig. 13. In a half random wafer, random defects fall in half of the wafer while the other half contains no pattern. A *Center+Edge-local* type wafer map contains an

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

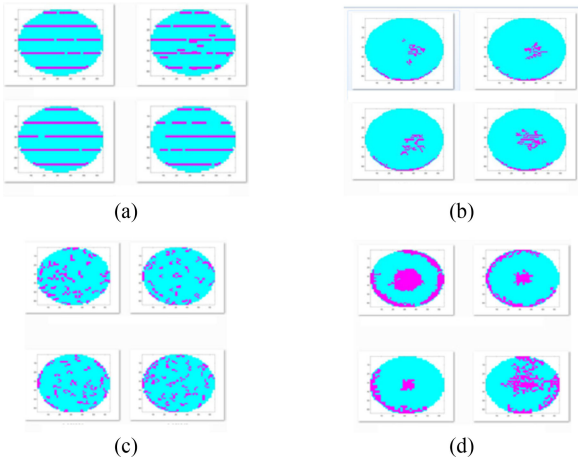


Fig. 12. Examples of four wafer map clusters defined by SOM.

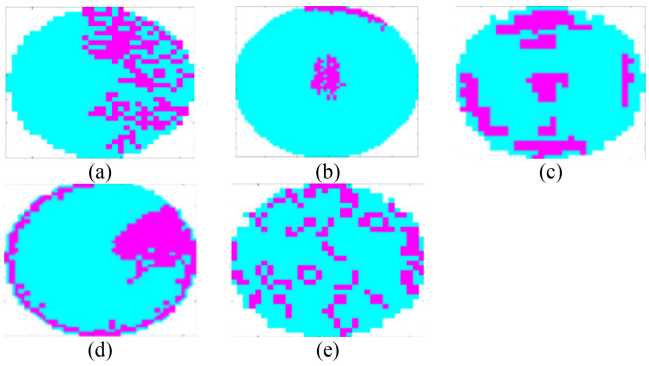


Fig. 13. (a) Half-random, (b) Center+Edge-local, (c) Center+Local, (d) Edge-ring+Local, (e) Normal.

edge pattern and a center defect pattern. The *Center+Local* type has a defect cluster at the center and some other defect clusters. An *Edge-ring+Local* type has an outer ring and a local cluster. Finally, there is no specific defect pattern in *Normal*, but the defect density is higher than *None* but lower than *Random*.

G. Model Verification

A complete iteration goes through the above six phases, as illustrated in Fig. 4. The labeling will be changed after an iteration, so the whole process has to be executed again. The labeling results will stabilize after several iterations. Once we have reached a stable labeling, the prediction model produced in Phase III of the last iteration is verified by using the labeled data to get the final results.

IV. Experimental Results

A. Experiment Setup

The proposed method is validated using the WK-811K dataset. The statistics of WK-811K are summarized in Fig. 14.

The dataset contains 811,457 wafer maps, in which roughly 20% of the samples are manually labeled. The dataset is extremely imbalanced, as 85.2% of the wafers belong to *None*. In contrast, four defect types contain relatively few wafers: *Donut* (0.3%), *Near-full* (0.1%), *Random* (0.5%), and *Scratch* (0.7%). Labeled samples in each defect type are divided into two parts: train and test. The sizes of train and test data given in Fig. 14 are designated by WK-811K. The distribution of the two parts varies from one type to another. Since we want to have enough training samples for each defect type, in this paper the labeled wafers are reorganized to ensure that the training set is slightly larger than the test set. The goal is to achieve stable classification results, especially for uncommon defect types. Let the number of wafers in a defect type be W_{total}, and the sizes of training data and test data be denoted as W_{train} and W_{test}. W_{train} is determined by W_{total} according to Table I, while W_{train} is equal to $W_{total} - W_{train}$.

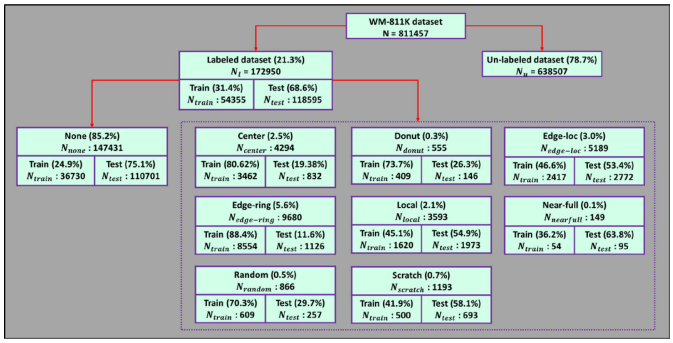


Fig. 14. Distribution of WM-811K dataset.

TABLE I
Parameter for Training Set Control

W_{total}	W_{train}
$W_{total} \leq 100$	$\lceil W_{total}/2 \rceil$
$100 < W_{total} \leq 200$	$\lceil W_{total}/2 \rceil$+20
$200 < W_{total} \leq 500$	$\lceil W_{total}/2 \rceil$+50
$500 < W_{total} \leq 1{,}000$	$\lceil W_{total}/2 \rceil$+100
$1{,}000 < W_{total} \leq 2{,}000$	$\lceil W_{total}/2 \rceil$+200
$2{,}000 < W_{total} \leq 10{,}000$	$\lceil W_{total}/2 \rceil$+500
$10{,}000 < W_{total} \leq 20{,}000$	$\lceil W_{total}/2 \rceil$+2,500
$20{,}000 < W_{total}$	20,000

B. Results

In the first iteration, the process starts with the training set of the labeled samples. At the end of each iteration, some unlabeled samples will be classified into a defect type, so the overall number of labeled will increase. In all, five iterations of the process are executed, as the amount of labeled data barely changed at the end of the fifth iteration. The change of labeled data in each iteration is illustrated in Fig. 15. Note that an initially unlabeled wafer map will be included only if it contains a defect pattern other than *None*. In other words, all remaining unlabeled data after Iteration 5 are actually classified as *None* by the proposed method.

The performance of a prediction model is usually evaluated using four metrics: precision, recall, F_1-score, and accuracy. Assume that N defect types are identified. Let TP_i, FP_i, TN_i, and FN_i are the numbers of true positives, false positives, true negatives, and false negatives for type i, $1 \leq i \leq N$,

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

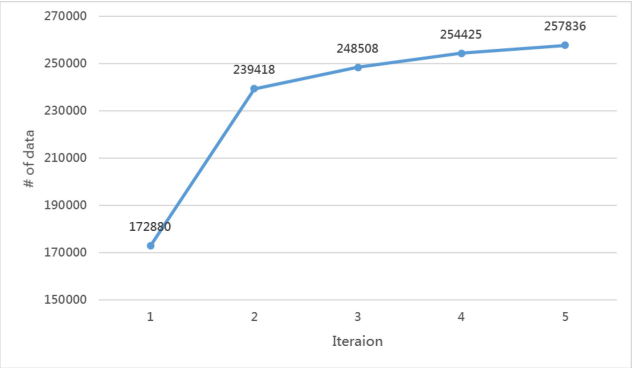


Fig. 15. The amount of labeled data at the beginning of each iteration.

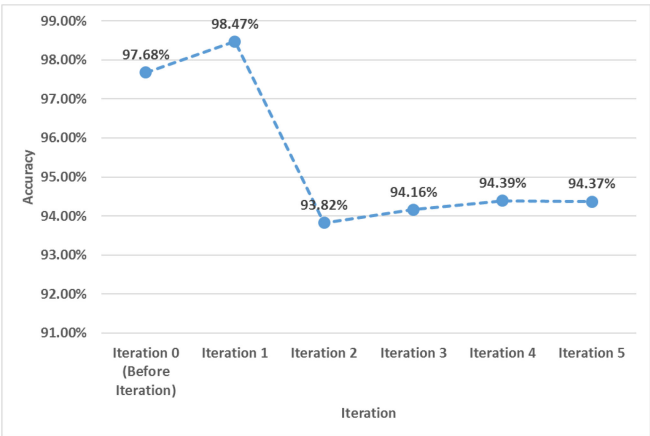


Fig. 16. Accuracy after each iteration.

respectively. The performance metrics are defined as follows.

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{1}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{2}$$

$$F_{1,i} = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{3}$$

$$Accuracy = \frac{\sum_{i=1}^{N} (TP_i + TN_i)}{\sum_{i=1}^{N} (TP_i + TN_i + FP_i + FN_i)} \tag{4}$$

The overall prediction accuracy achieved at the end of each iteration is given in Fig. 16. Iteration 0 is provided for comparison, and the prediction model is constructed by applying the training data directly and then the model is evaluated by the test data. The proposed method can improve the classification results, as the overall accuracy improves after Iteration 1. The accuracy drop between Iterations 1 and 2 will be explained later. For Iterations 1 to 5, the results are obtained through the process discussed in the previous section.

It seems that the accuracy of Iteration 0 is good enough (97.68%); however, a closer look at the data suggests otherwise. The recall rates in 4 out of the 8 real defect types are lower than 80%. The high accuracy is skewed by the large amount of wafers without defect patterns, as more than 85.5% wafers below to *None*, and the recall rate for *None* is extremely high (99.18%). The recall rates of the 9 defect patterns defined in WM-811K [8] are illustrated in Fig. 17 for all iterations. It can be seen that the recall rates for these defect patterns are relatively close among all iterations except in Iteration 0, in which the proposed process has not yet been applied.

The classification results after Iteration 1 give the effectiveness of the proposed supervised learning flow for the labeled data in WM-811K. With the same set of data, the proposed flow significantly improve the recognition rates (i.e., recall values) for all the real defect patterns, as shown in Fig. 17. As a result, the accuracy increases to 98.47% in Iteration 1.

TABLE II
PERFORMANCE OF THE PROPOSED METHOD AFTER THE 1st AND 5th ITERATIONS

Type	Iteration	#Wafers	Precision	Recall	F_1
Center	I1	4292	96.41%	94.84%	95.62%
	I5	7,999	94.63%	90.22%	92.37%
Center+ Edge-local	I1	-	-	-	-
	I5	9,633	87.97%	91.89%	89.89%
Center+ Local	I1	-	-	-	-
	I5	4,658	80.59%	96.03%	87.64%
Donut	I1	555	92.79%	95.90%	94.32%
	I5	849	89.28%	95.71%	92.38%
Edge-local	I1	5,177	87.85%	90.13%	88.98%
	I5	29,814	94.87%	88.60%	91.63%
Ring	I1	9,680	98.61%	98.62%	98.61%
	I5	10,531	96.14%	99.02%	97.56%
Half-random	I1	-	-	-	-
	I5	35	51.43%	94.74%	66.67%
Local	I1	3,587	85.48%	89.28%	87.34%
	I5	6,982	85.69%	89.73%	87.66%
Near-full	I1	149	95.30%	100.00%	97.59%
	I5	424	96.93%	98.33%	97.62%
Normal	I1	-	-	--	--
	I5	25,651	94.49%	81.92%	87.76%
Random	I1	865	93.87%	95.08%	94.47%
	I5	10,437	91.51%	92.02%	91.76%
Ring+ Local	I1	-	-	-	-
	I5	171	52.05%	92.71%	66.67%
Scratch	I1	1,190	76.72%	96.00%	85.29%
	I5	3,908	86.00%	87.80%	86.89%
None	I1	147,385	99.44%	99.11%	99.28%
	I5	146,724	95.99%	98.81%	97.38%
Total	I1	172,880			
	I5	257,836			

TABLE III
CLASSIFYING UNLABELED DATA USING MODELS CONSTRUCTED IN THE 1st AND 5th ITERATIONS

	Iteration 1		Iteration 5	
	#Wafers	Ratio	#Wafers	Ratio
Center	20,445	3.28%	12,527	2.01%
Center+Edge-local	-	-	15,734	2.52%
Center+Local	-	-	3,969	0.64%
Donut	235	0.04%	461	0.07%
Edge-local	27,346	4.38%	51,454	8.25%
Ring	9,840	1.58%	5,208	0.83%
Half-random	-	-	17	0.00%
Local	7,714	1.24%	5,803	0.93%
Near-full	886	0.14%	2,217	0.36%
Normal	-	-	45,869	7.35%
Random	7,295	1.17%	14,765	2.37%
Ring+Local	-	-	80	0.01%
Scratch	12,964	2.08%	3,801	0.61%
None	537,198	86.10%	462,018	74.05%
Total	623,923	100%	623,923	100%

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
COMPARISON OF MACHINE LEARNING METHODS

	Center	Center+ Ed-ge-local	Center+ Local	Donut	Edge-Local	Ring	Half-random	Local	Near-full	Normal	Random	Ring+ Local	Scratch	None	Acc.
M-SVM [9]	84.90%	-	-	74.00%	85.10%	79.70%	-	68.50%	98.90%	-	79.80%	-	82.40%	95.70%	94.63%
JLNDA [2]	93.00%	-	-	98.00%	77.00%	**100.00%**	-	81.00%	-	-	94.00%	-	-	-	90.50%
OPTICS[10]	95.60%	-	-	77.50%	92.30%	98.40%	-	81.80%	**100.00%**	-	85.80%	-	74.40%	-	94.30%
D-Tree [11]	95.75%	-	-	92.25%	83.50%	86.75%	-	83.50%	-	-	95.75%	-	86.00%	**100.00%**	90.50%
ESDAE [13]	92.90%	-	-	88.80%	**94.10%**	94.30%	-	68.20%	52.70%	-	79.60%	-	87.70%	-	89.50%
CNN [14]	91.00%	-	-	86.00%	82.00%	**100.00%**	-	88.00%	**100.00%**	-	95.00%	-	93.00%	99.98%	93.13%
DCNN [15]	**98.00%**	-	-	**98.30%**	93.10%	91.70%	-	**90.30%**	99.90%	-	**96.50%**	-	**98.70%**	99.70%	96.20%
LWNN [16]	92.65%	-	-	83.00%	84.18%	98.11%	-	79.25%	93.48%	-	83.84%	-	77.00%	98.05%	93.95%
Proposed-I1	94.84%	-	-	95.90%	90.13%	98.62%	-	89.28%	**100.00%**	-	95.08%	-	96.00%	99.11%	98.47%
Proposed-I5	90.22%	**91.89%**	**96.03%**	95.71%	88.60%	99.02%	**94.74%**	89.73%	98.33%	**81.92%**	92.02%	**92.71%**	87.80%	98.81%	94.37%

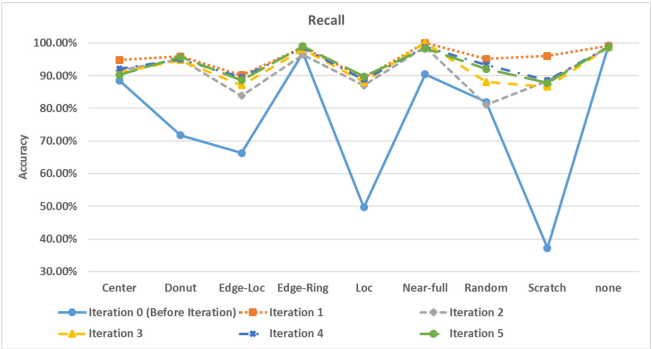


Fig. 17. Recall values of the WM-811K defect patterns.

A significant drop of the overall accuracy occurs between Iterations 1 and 2 (from 98.47% to 93.82%) and the drop is attributed to two reasons. First, three new labels are created after Iteration 1 (*Center+Edge-local*, *Center+Local*, *Normal*), and the numbers of wafer maps in these three types are relatively large. The creation of new types affects the recall rates of the other existing types (see Fig. 17), and the recall rates of the new types are not good in the first place. Second, a lot of unlabeled data have been classified and labeled in Iteration 2. However, wafers without real defect patterns (i.e., *None*) are not added to the labeled data, as discussed before. In other words, the ratio of *None* will drop in each iteration. Since the recall rate of *None* is very close to 100%, the overall accuracy will be adversely affected by the decreasing ratio of *None*.

The other two new labels are created after Iteration 4 (*Half-random*, *Ring+Local*). Similar to the impact of new labels in Iteration 2, the overall accuracy drops slightly in Iteration 5 (Fig. 16), and the recall rates of existing types are slightly lower as well (Fig. 17). The process stops after Iteration 5 due to the following two facts: (1) the numbers of wafers in the new types are extremely small (see Table II), and (2) the impact on the accuracy is minimal.

The performance metrics of the proposed method after the first and fifth iterations are summarized in Table II, and the number of wafers in each defect type is given as well. Most of the defect types can be recognized with good results, as the F_1-scores are higher than 85% except for *Half-random* and *Ring+Local*. In each of the two cases, the recall is good but the precision is not. The reason is mainly due to the rather small number of samples in these two cases.

The prediction models constructed in the first and fifth iterations are used to classify the unlabeled data, and the results are reported in Table III. Note that the model constructed in the first iteration is trained using the labeled data in the WM-811K dataset, so only nine defect patterns are classified. In this case, the ratio of *None* is 86.10%, which is consistent with the labeled data. On the other hand, the ratio of *None* type wafers reduces to 74.05% after Iteration 5, and the distribution of defect types is somewhat different. The results indicate that the original 9 types are likely not enough, and the new defect types defined through unsupervised learning can further improve the classification resolution.

A comparison of the proposed method with previous methods is given in Table IV. The results show that the proposed method achieves comparable recall values for all defect types. Only [15] consistently gives good classification results for all defect types, but the results are achieved through augmented data set so that all defect types have relatively balanced training data. In contrast, the proposed methods only apply the WM-811K wafer maps, which are inherently imbalanced.

The most time-consuming part of the proposed method is Phase III, as 10 iterations of the Bagging algorithm is executed and a large number of decision trees (1,000) are used for ensemble learning. The reason is that we want to set a more precise threshold for error wafer detection. Furthermore, the process is not fully automated and the WM-811K database is very large. In general, it takes several days to complete an iteration on a PC with Intel Core i5-8365U CPU @ 1.6 GHz and 16 GB DRAM. It is possible to streamline the process by selecting a smaller Bagging model that achieves acceptable performance.

V. CONCLUSION

Supervised method learning algorithms provide a promising direction to realize automatic wafer defect maps recognition. The effectiveness of a ML prediction model is affected by the training samples. However, the manual labeling process is subjective, and we have seen examples of assigning

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

similar defect patterns to different types. Furthermore, some defect patterns are not defined in the first place. In this paper, we propose a semi-supervised framework to deal with the problems. An ensemble learning algorithm is used for supervised learning and identify possibly mislabeled samples, and unsupervised learning is applied to extracted undefined defect types. The proposed method can be used to improve labeling to achieve better recognition rates for wafer defect patterns.

REFERENCES

[1] S.-C. Hsu and C.-F. Chien, "Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing," *Int. J. Prod. Econ.*, vol. 107, no. 1, pp. 88–103, May 2007, doi: 10.1016/j.ijpe.2006.05.015.

[2] J. Yu and X. Lu, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 1, pp. 33–43, Feb. 2016, doi: 10.1109/TSM.2015.2497264.

[3] F.-L. Chen and S.-F. Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 3, pp. 366–373, Aug. 2000, doi: 10.1109/66.857947.

[4] C.-F. Chien, S.-C. Hsu, and Y.-J. Chen, "A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence," *Int. J. Prod. Res.*, vol. 51, no. 8, pp. 2324–2338, Apr. 2013, doi: 10.1080/00207543.2012.737943.

[5] L. Mika and Y. Hiltunen, "Recognition of systematic spatial patterns in silicon wafers based on SOM and K-means," *IFAC PapersOnLine*, vol. 51, no. 2, pp. 439–444, 2018, doi: 10.1016/j.ifacol.2018.03.075.

[6] S. Lee and D. Kim, "Distributed-based hierarchical clustering system for large-scale semiconductor wafers," in *Proc. Int. Conf. Ind. Eng. Eng. Manag. (IEEM)*, Dec. 2018, pp. 1528–1532, doi: 10.1109/IEEM.2018.8607492.

[7] S. J. Bae, J. Y. Hwang, and W. Kuo, "Yield prediction via spatial modeling of clustered defect counts across a wafer map," *IIE Trans.*, vol. 39, no. 12, pp. 1073–1083, Dec. 2007, doi: 10.1080/07408170701275335.

[8] F. Adly, P. D. Yoo, S. Muhaidat, Y. Al-Hammadi, U. Lee, and M. Ismail, "Randomized general regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 2, pp. 145–152, May 2015, doi: 10.1109/TSM.2015.2405252.

[9] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015, doi: 10.1109/TSM.2014.2364237.

[10] M. Fan, Q. Wang, and B. V. D. Waal, "Wafer defect patterns recognition based on OPTICS and multi-label classification," in *Proc. Adv. Inf. Manag. Commun. Electron. Autom. Control Conf.*, 2017, pp. 912–915, doi: 10.1109/IMCEC.2016.7867343.

[11] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 250–257, May 2018, doi: 10.1109/TSM.2018.2806931.

[12] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018, doi: 10.1109/TSM.2018.2795466.

[13] J. Yu, "Enhanced stacked denoising autoencoder-based feature Learning for recognition of wafer map defects," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 613–624, Nov. 2019, doi: 10.1109/TSM.2019.2940334.

[14] N. Yu, Q. Xu, and H. Wang, "Wafer defect pattern recognition and analysis based on convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 566–573, Nov. 2019, doi: 10.1109/TSM.2019.2937793.

[15] M. Saqlain, Q. Abbas, and J. Y. Lee, "A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 436–444, Aug. 2020, doi: 10.1109/TSM.2020.2994357.

[16] T.-H. Tsai and Y.-C. Lee, "A light-weight neural network for wafer map classification based on data augmentation," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 4, pp. 663–672, Nov. 2020, doi: 10.1109/TSM.2020.3013004.

[17] L.-Y. Chen *et al.*, "Semi-supervised framework for wafer defect pattern recognition with enhanced labeling," in *Proc. IEEE Int. Test Conf. (ITC)*, 2021, pp. 208–212, doi: 10.1109/ITC50571.2021.00029.

[18] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Disc. Data Min.*, 1996, pp. 226–231, doi: 10.5120/739-1038.

[19] "Image moment." [Online]. Available: https://en.wikipedia.org/wiki/Image_moment (Accessed: Feb. 2022).

[20] L.-Y. Chen *et al.*, "TestDNA-E: Wafer defect signature for pattern recognition by ensemble learning," in *Proc. ITC*, 2020, pp. 1–4, doi: 10.1109/ITC44778.2020.9325237.

[21] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Harlow, U.K.: Prentice-Hall, 2008.

[22] P. N. Tan, M. Steinbach, and V. Kumar, "Methods for constructing an ensemble classifier," in *Introduction to Data Mining*. Boston, MA, USA: Pearson Educ., 2006, ch. 5, pp. 276–280.

[23] T. Kohonen, *Self-Organizing Maps*, 3rd ed. New York, NY, USA: Springer, 2001.

[24] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 901–909.

[25] L. Shapiro and G. Stockman, *Computer-Vision*. Upper Saddle River, NJ, USA: Prentic Hall, 2001, ch. 3, pp. 69–73.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 20,2023 at 18:09:28 UTC from IEEE Xplore. Restrictions apply.

Semi-Supervised Framework for Wafer Defect Pattern Recognition with Enhanced Labeling

Leon Li-Yang Chen
NXP Semiconductors Taiwan Ltd.
Kaohsiung, Taiwan
leon.chen@nxp.com

Katherine Shu-Min Li
Dept Computer Science and Eng.
National Sun Yat-Sen University
Kaohsiung, Taiwan
smli@cse.nsysu.edu.tw

Xu-Hao Jiang
Dept Computer Science and Eng.
National Sun Yat-Sen University
Kaohsiung, Taiwan
tyst89033@yahoo.com.tw

Sying-Jyan Wang
Dept Computer Science and Eng.
National Chung Hsing University
Taichung, Taiwan
sjwang@cs.nchu.edu.tw

Andrew Yi-Ann Huang
NXP Semiconductors Taiwan Ltd.
Kaohsiung, Taiwan
andrew.huang@nxp.com

Jwu E Chen
Dept. Electrical Eng.
National Central University
Chungli, Taiwan
jechen@ee.ncu.edu.tw

Hsing-Chung Liang
Dept. Electronics Engineering
Chung Yuan Christian University
Chungli, Taiwan
hcliang@cycu.edu.tw

Chun-Lung Hsu
Industrial Technology Research Institute
Hsin Chu, Taiwan
clh@itri.org.tw

2021 IEEE International Test Conference (ITC) | 978-1-6654-1695-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITC50571.2021.00029

***Abstract*—Wafer map defect pattern recognition is valuable for root cause analysis and yield learning. Most of the previous studies on defect pattern recognition are based on supervised machine learning, in which labeled wafer maps are used to train a machine learning model for automatic classification. Some problems arise in this approach. First, there may be misclassification in the original labeled data, which makes it difficult to establish an accurate prediction model. Secondly, defect patterns that are not defined before will not be classified correctly. In this paper, we proposed a semi-supervised framework to deal with these problems. Labeled wafer maps are first used to train a prediction model, with likely misclassified data excluded. The prediction model is then used to classify unlabeled data. The remaining data that cannot be properly classified are then sent to an unsupervised learning algorithm to extract more defect patterns with enhanced labeling techniques. This proposed approach is validated with TSMC 811K database, in which we are able to define five new defect pattern types. Experimental results show that total 14 defect types can be recognized with overall accuracy of 94.37%.**

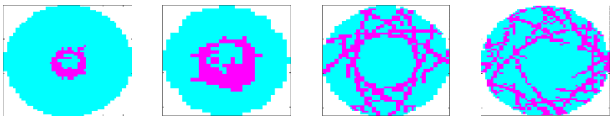
Keywords—wafer map, defect pattern, yield learning, unsupervised machine learning, machine learning, enhanced labeling.

I. Introduction

Integration circuit (IC) fabrication is very complicated, as many manufacturing steps have to be executed on the same wafer. All fabrication steps are subject to errors, wafer defect patterns are the results of various problems in the fabrication and wafer test process. Thus, the wafer map defect patterns can be used to identify sources of errors in the manufacturing process. Recently, Local defect pattern recognition has attracted a lot of research interests [1]-[6]. The results of wafer test are usually presented in a wafer map, which gives locations of failed dies in a wafer.

State-of-art research on wafer defect pattern recognition starts with the work by Wu, Jang, and Chang in 2015 [1] using the TSMC 811K database (WM-811K). This work provides a supervised machine learning framework for defect pattern recognition. Nine defect types are identified, in which seven types can be identified with higher than 80% recognition rate. Roughly 95% of the labeled wafers are classified as non-pattern, as no spatial patterns are found in such wafers. Only 21.3% of the WM-811K wafer maps are labeled, while the majority (78.7%) wafer maps are unlabeled.

Previous defect pattern recognition methods can achieve relatively good but not great classification accuracy, and it can be attributed to three reasons. The first reason is incorrect labeling due to human errors. The labeling process is subjective, and sometimes the same wafer map can be assigned to two or even more different types by the same person at different times. Some examples of incorrect labeling are given in Fig. 1. Fig. 1(a) and Fig. 1(b) are similar, but one is classified as center while the other is doughnut. Both Fig. 1(c) and Fig. 1 (d) should be scratch type, but Fig. 1(c) is classified as doughnut instead. That is, Fig. 1(a) and Fig. 1(c) show the manual mislabeling, which leads to the challenge in supervised learning.



(a) (b) (c) (d)

Fig. 1. Example of a figure caption. (*figure caption*)

The second motivation is that some wafer maps contain multiple patterns, but the labeling is usually carried out according to a given priority order. The existence of such wafers in the training set may create an skewed model, which leads to misclassifications. The last reason is inadequate samples in some defect types. In order to deal with the aforementioned problems, we have developed a semi-supervised learning method for the automatic classification of wafer maps. We have applied the method to the entire WM-811K database, including the unlabeled wafer maps.

Experimental results show that total 14 defect types with 5 new defect types can be recognized with overall accuracy of 94.37%.

IP Paper

978-1-6654-1695-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ITC50571.2021.00029

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:27:24 UTC from IEEE Xplore. Restrictions apply.

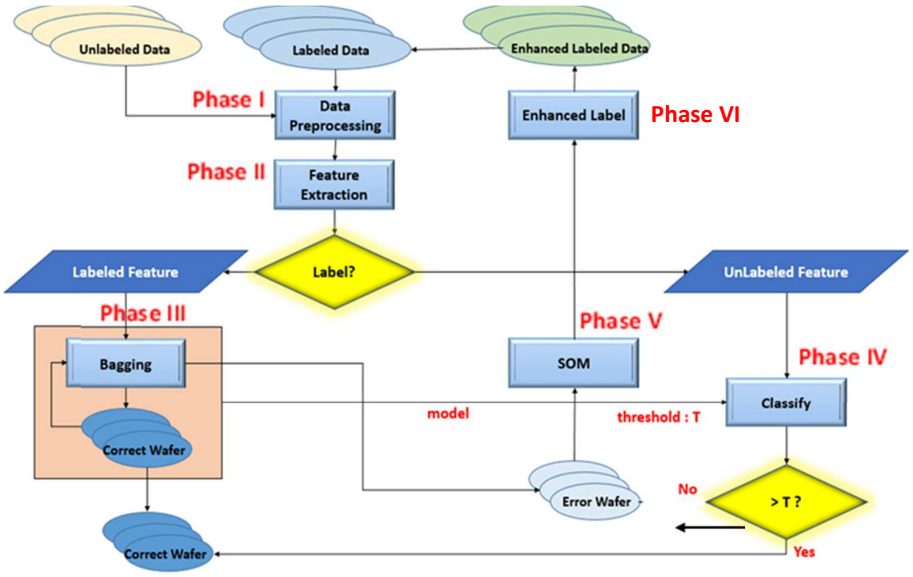


Fig. 2. Semi-supervised framework of wafermap pattern recognition with enhanced labeling.

II. Semi-Supervised with Enhanced Labeling Framework for Wafermap Identification

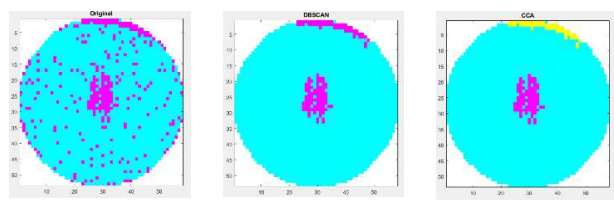
A. Overall Framework

The overall flow in the proposed framework is illustrated in Fig. 2. The input database consists of labeled and unlabeled wafer maps. All wafers are preprocessed in Phases I and II such that useful features are extracted for the following machine learning algorithms, and then the labeled wafer maps are used to build a prediction model in Phase III. However, since there are *mislabeled* and/or *ambiguous labeling* in the database, some wafer maps may not fit in the model. The prediction model constructed in Phase III is then used to classify the unlabeled wafer maps in Phase IV. Wafers that can be classified with enough confidence (larger than threshold *T*) are listed as "correct wafers", while the others are also "error wafers". We will then apply an unsupervised learning technique called self-organizing map (SOM) to classify the error wafers and mislabeled wafers in Phase V. Additional classes are constructed to label the clustered wafers in Phase VI. The enhanced labeled data are combined with the correct wafers to form the new labeled data. The process will be repeated through several iterations.

B. Phase I: Preprocessing

Three tasks are carried out in this phase. First, some items in WM-811K are not specified properly. In all, roughly 1.84% of the wafer maps are discarded due to erroneous format or missing data in the wafer maps.

The second task is to remove random defects in a wafer map such that a clean pattern can be extracted. An example is shown in Fig. 2(a), which contains two local defect patterns and some randomly distributed defects. The task is achieved by using DBSCAN to retrieve clusters in a map. Fig. 2(b) shows the result of denoising.



(a) (b) (c)

Fig. 3. (a) Original wafer map, (b) after denoising with DBSCAN, (c) CCA.

The wafer maps in WM-811K are given in various sizes. To facilitate machine learning, the wafers are transformed into four standard sizes: 32×32, 64×64, 128×128, and 256×256.

C. Feature Extraction

Features that will be used by the supervised and unsupervised machine learning algorithms are extracted in this phase. Three categories of features are used.

The first category is the geometric features. First, we need to locate clusters of defects in a map, and this is achieved through connected-component analysis (CCA). For example, two connected components can be found in Fig. 3(c). The geometric features include the number of connected components and geometric information related to the two largest components in the wafer. For each component, the following parameters are used: (1) distance between component center and wafer center, (2) distance from component center to wafer edge, (3) area, (4) eccentricity, and (5) solidity.

The second category is the statistic features. We employ four classes of statistic features. The first class is related the entire wafer, and the feature is the ratio of defective dies in the whole wafer. The second class contains information about individual components. First we calculate the size of each component vs. all defective dies. Then, the mean, variance, maximum and minimum of the ratios are used as the features in the second class.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:27:24 UTC from IEEE Xplore. Restrictions apply.

The third class is the image moment [7], also known as Hu's invariant [7]. The fourth class contains defect density (DD) and defect clustering (DC) in three areas: edge, ring, and local [8].

The third category is related to Radon Transform of the wafer map. Once we have the transformed matrix, we can use the maximum, mean, and variance of the row vectors as the Radon features. Fig. 4 gives the maximum, mean, and variance of 20 row vectors for nine defect patterns, and it can be seen that such features are very useful in terms of distinguishing these patterns.

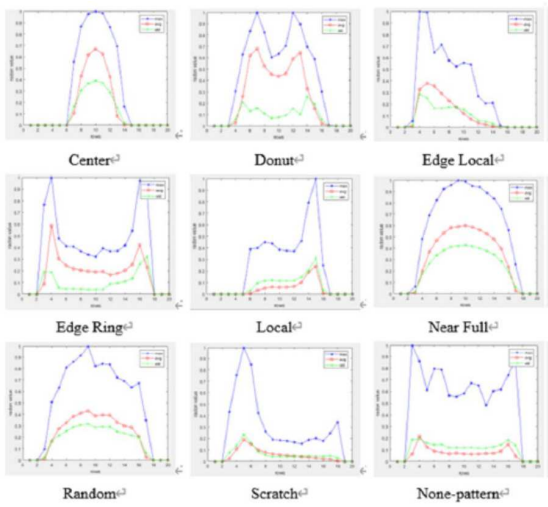


Fig. 4. Maximum, mean, and variance of the 20 row vectors of Radon tranform matrices.

D. Phase III: Classification Model Construction

The goal of Phase III is to build a low-tolerance classification model according to labeled data that are currently available. Since the tolerance threshold is very low, misclassified data in WM-811K can also be screened in this step.

The Bootstrap aggregating, also called Bagging algorithm, is used for supervised learning. We employ 1,000 decision trees as the sub-classifiers, and the final classification is determined according to the results of the sub-classifiers. A sample is classified as defect type d if the largest number of sub-classifiers as type d. Assume k sub-classifiers assigned the sample to type d, then the probability of the sample is $k/1000$.

The prediction model is then used to classify all the labeled data to screen out misclassified data as error wafers. The remaining data are then applied to Bagging again to build a new model. The final model is built after 10 iterations of the training process.

The confidence threshold T is obtained from the final model as follows. (1) Every sample is classified by the model, and probability that the sample is assigned to the corresponding defect type is recorded. (2) The standard deviation σ of these probability values is calculated. (3) Let the average of the probabilities for all samples classified to defect type d be $u(P_d)$. The threshold of confidence level for defect type d is then set to $u(P_d)+/-2\sigma$, which implies confidence interval is 95%.

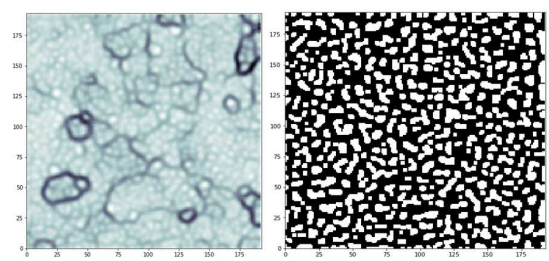
E. Phase IV: Classifying Unlabeled Data

The prediction model built in Phase III is used to classify unlabeled data in this phase. An unlabeled wafer map will be assigned to type d if its probability is higher the threshold obtained in the previous phase. If a wafer map cannot be associated with any known defect type, it is classified as "error wafer".

F. Phase V: Unsupervised Learning

In this phase, we will cluster similar wafer maps, which are not currently labeled into defined defect types, into similar groups. The task is carried out by self-organized map (SOM), an artificial neural network (ANN) model trained using unsupervised learning. For a set of input data, an SOM can produce a low-dimensional map that preserves the topological properties of the input space. This makes SOMs useful for visualization of the input data.

The procedure of SOM is outlined as follows. First, the features obtained in Phase II are represented by a characteristic matrix W of size $m \times n$, where m and n are the number of unlabeled wafers and the number of features, respectively. The matrix is normalized by multiway normalization [9]. The normalized matrix is then used to train the SOM, and the result is the U-matrix shown in Fig. 5(a). The grayscale gives the difference among weights of neighboring neurons, with darker shades meaning larger differences. It can be seen that the differences among neighboring clusters are not clear enough at this stage.



(a) (b)

Fig. 5. Unsupervised learning by SOM, (a) U-matrix, (b) enhanced walls.

In order to clearly distinguish different clusters, we apply two techniques to enhance the walls that separate white areas. The first technique is the *median wall finding*, which is used to determine the locations of walls. The second one is the *closing technique* developed in image processing. In general, closing is used to remove holes between walls to create clear boundary among white areas. The results of *median wall finding,* or *enhanced walls,* is shown in Fig. 5(b), Finally, when the unlabeled data are superimposed to enhanced U-matrix, as shown in Fig. 6, they are partitioned into groups of wafer maps that are classified as the same defect type. If a data point falls in the area of a wall, the data point will be included in a neighboring class with the shortest Euclidean distance. In Fig. 6, It can be seen that the enhanced U-matrix provides an efficient way to classify wafer maps. Still, in some areas, there are points of different colors mixed in the same region. In reality, it means that the supervised model is unstable in some types, which may be caused by insufficient training data or the existence of wrong

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:27:24 UTC from IEEE Xplore. Restrictions apply.

labeling, which makes the enhanced labeling in the next phase necessary.

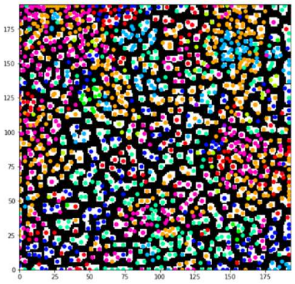


Fig. 6. Results of unsupervised learning with SOM clustering.

G. Phase VI: Enhanced Labeling

The new SOM groups need to be labeled by human. Some groups are very similar to existing defect pattern types, so they will be incorporated into the existing types. Otherwise, new defect labels will created for the new defect groups, and the wafer defect database is updated accordingly.

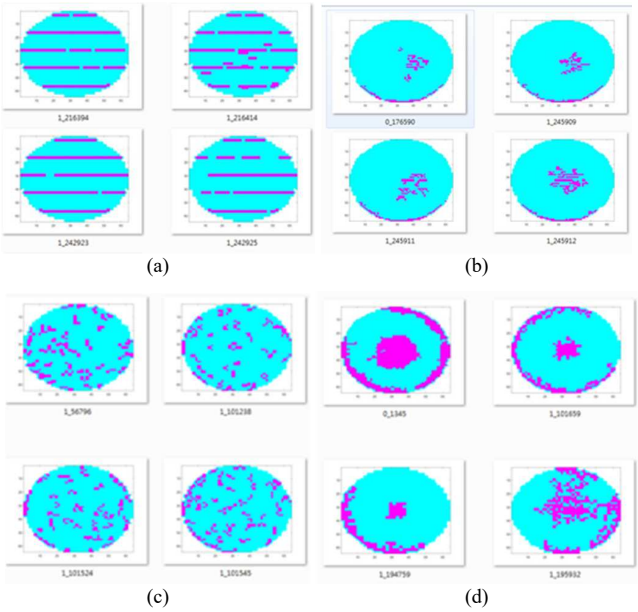


Fig. 7. Examples of four wafer map clusters defined by SOM.

In Fig. 8, we define four new defect pattern types in WM-811K: Half Random, Center-EdgeLocal, Center-Local, and Ring-Local. The first one (Half Random) consists of a simple pattern, while the other three wafer maps consist of two defect patterns. Examples of these defect pattern types are given in Fig. 8. In a half random wafer, random defects fall in half of the wafer while the other half contains no pattern. A Center-EdgeLocal type wafer map contains an edge pattern and a local defect pattern. The Center-Local type has a defect cluster at the center and some other defect clusters. A Ring-Local type has an outer ring and a local cluster. The proposed method is validated using TSMC WM-811K database. The wafer maps are partitioned into four groups according to wafer map size. Wafer maps in each group are classified by the proposed method, and the results are given in the four confusion matrices given in Figures 9 to 12. Compared with previous methods, Table I shows that the proposed method achieves 94.37% classification accuracy even though 14 defect pattern types are defined in our work.

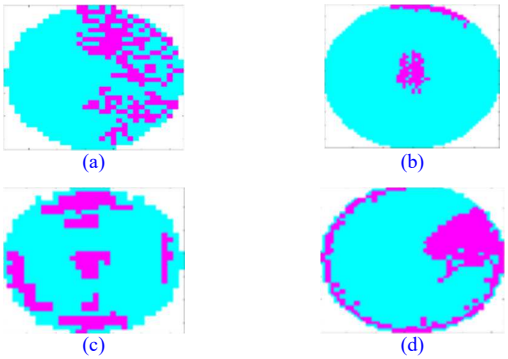


Fig. 8. (a) Half random, (b) Center-EdgeLocal, (c) Center-Local, (d) Ring-Local. Experimental Results

REFERENCES

[1] M. J. Wu, J. S. R. Jang, and J. L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1-12, Feb. 2015.

[2] F. Adly , O. Alhussein, P. D. Yoo, Y. Al-Hammadi, K. Taha, S. Muhaidat, Y.-S. Jeong, U. Lee, and M. Ismail, "Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Industrial Informatics*, vol. 11, no. 6, pp. 1267-1276, Dec. 2015.

[3] J. Yu and X. Lu, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 1, pp. 33-43, Feb. 2016.

[4] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 250-257, May 2018.

[5] R. Wang and N. Chen, "Wafer map defect pattern recognition using rotation-invariant features," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 596-604, Nov. 2019.

[6] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309-314, May 2018.

[7] Available: https://en.wikipedia.org/wiki/Image_moment.

[8] L.L.-Y. Chen, et. al, "TestDNA-E: wafer defect signature for pattern recognition by ensemble learning," in Proc. ITC, 2020, pp. 1-4.

[9] T. Salimans and D. P. Kingma, "Weight normalization: a simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst*., 2016, pp. 901–909.

[10] M. Fan, Q. Wang, and B. van der Waal, "Wafer defect patterns recognition based on OPTICS and multi-label classification," in Proc. 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conf. (IMCEC) , 2016, pp. 912-915.

[11] J. Yu, "Enhanced stacked denoising autoencoder-based feature Learning for recognition of wafer map defects," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 613-624, Nov. 2019.

[12] N. Yu, Q. Xu, and H. Wang, "Wafer defect pattern recognition and analysis based on convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 566-573, Nov. 2019.

[13] M. Saqlain, Q. Abbas, and J. Y. Lee, "A deep convolutional neural network for wafer defect identification on an imbalanced dataset in

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:27:24 UTC from IEEE Xplore. Restrictions apply.

semiconductor manufacturing Processes," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 436-444, Aug. 2020.

[14] T. -H. Tsai and Y. -C. Lee, "A light-weight neural network for wafer map classification based on data augmentation," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 4, pp. 663-672, Nov. 2020.

TABLE I. COMPARISON OF MACHINE LEARNING METHODS

	Center	Center-Edge-Loc	Center-Loc	Donut	Edge-Loc	Edge-Ring	Half-Random	Loc	NearFull	Normal	Random	Ring-Loc	Scratch	none	Acc.
M-SVM [1]	84.90%	NA	NA	74.00%	85.10%	79.70%	NA	68.50%	98.90%	NA	79.80%	NA	82.40%	95.70%	94.63%
JLNDA [3]	93.00%	NA	NA	98.00%	77.00%	**100.00%**	NA	81.00%	NA	NA	94.00%	NA	NA	NA	90.50%
OPTICS[10]	95.60%	NA	NA	77.50%	92.30%	98.40%	NA	81.80%	**100.00%**	NA	85.80%	NA	74.40%	NA	94.30%
D-Tree [4]	95.75%	NA	NA	92.25%	83.50%	86.75%	NA	83.50%	NA	NA	95.75%	NA	86.00%	**100.00%**	90.50%
ESDAE [11]	92.90%	NA	NA	88.80%	94.10%	94.30%	NA	68.20%	52.70%	NA	79.60%	NA	87.70%	NA	89.50%
CNN [12]	91.00%	NA	NA	86.00%	82.00%	**100.00%**	NA	88.00%	**100.00%**	NA	95.00%	NA	93.00%	99.98%	93.13%
DCNN [13]	**98.00%**	NA	NA	**98.30%**	93.10%	91.70%	NA	90.30%	99.90%	NA	**96.50%**	NA	**98.70%**	99.70%	**96.20%**
LWNN [14]	92.65%	NA	NA	83.00%	84.18%	98.11%	NA	79.25%	93.48%	NA	83.84%	NA	77.00%	98.05%	93.95%
Proposed	90.22%	**91.89%**	**96.03%**	95.71%	88.60%	99.02%	**94.74%**	89.73%	98.33%	81.92%	92.02%	92.71%	87.80%	98.81%	94.37%

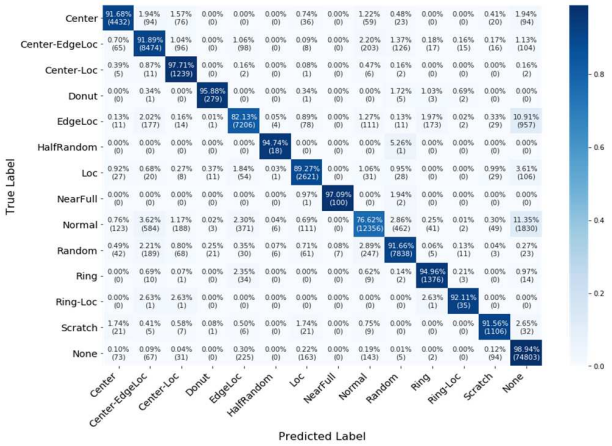


Fig. 9. Confusion matrix of the 32×32 model.

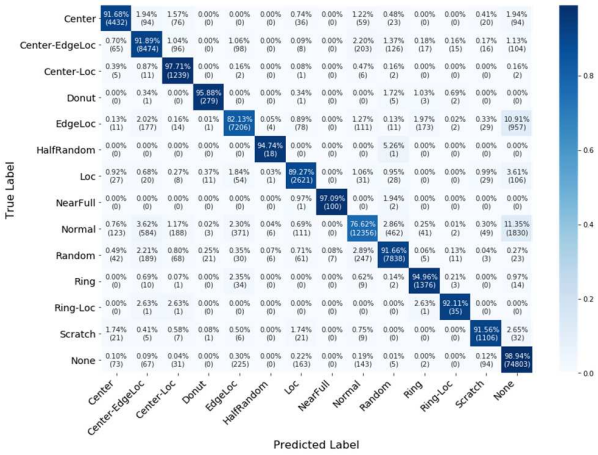


Fig. 11. Confusion matrix of the 128×128 model.

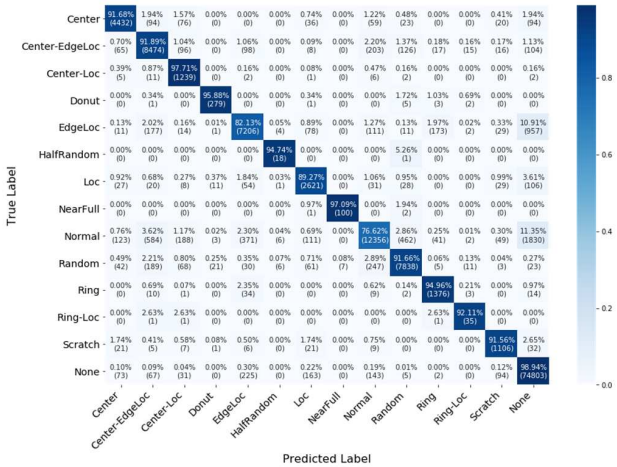


Fig. 10. Confusion matrix of the 64×64 model.

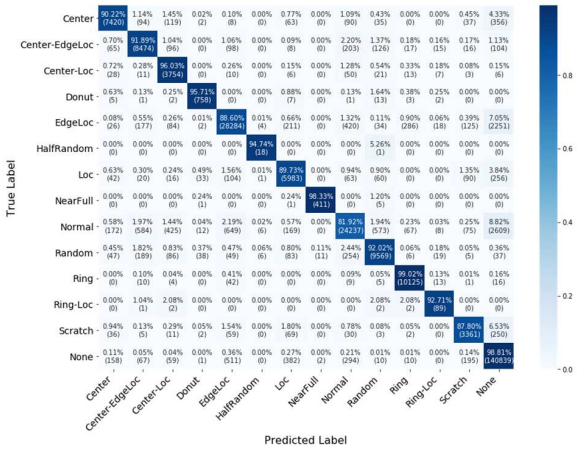


Fig. 12 Confusion matrix of the 256×25 mo

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:27:24 UTC from IEEE Xplore. Restrictions apply.

Improved Color Defect Detection With Machine Learning for After Develop Inspections in Lithography

Matthew P. McLaughlin, Petra Mennell, Andrew Stamper, Gabriel Barber, Janice Paduano, Emerson Benn, Michael Linnane, Justin Zwick, Chetan Khatumria, Robert L. Isaacson, Nathan Hoffman, and Clayton Menser

***Abstract*—Color defect detection was improved with the development of a Machine Learning (ML) method for after develop inspections in lithography. For coating defects, the method exhibited two times the sensitivity and three times the specificity in a trial comparison against the reference method. Using the ML method for disposition along with recipe optimization, predictive maintenance, and rework for coating defects, reduced yield loss from splatters in the long run by over 20x. Herein we describe learnings on the use of image enhancement for training and disposition, an Explainable AI application to support understanding, and a process flow to train augmentation based on performance.**

***Index Terms*—Machine learning, artificial intelligence, cloud manufacturing, defect detection, image enhancement, resist splatter, yield impact, after develop inspection, lithography, semiconductors manufacturing.**

I. Introduction

SEMICONDUCTOR manufacturing sites, also commonly called Fabs, employ a myriad of ways to control the quality of the wafers while they are processing through the manufacturing facility. As the current global chip shortage has put a spotlight on semiconductor chips, this article describes a way to improve output or yield at wafer final test by effectively detecting and eradicating a specific defect early on. The defect treated in this work is a defect that can affect all products made inside a Fab. Detection and eradication of this defect relies on three factors: 1. sensitive wafer final test of every die of the still whole wafer; 2. sufficiently small die size, as well as 3. sufficiently sensitive inline control of every wafer. All three of these factors applied in this current study, allowing accurate tracking of the resulting yield improvement of the final wafers.

Wafer final test provides pass-fail for each die for a range of parameters being tested to evaluate functionality of each chip. [1] Die size can vary and range from centimeters to less than a millimeter in size, depending on desired chip function and content. Certain large size defects with an irregular shape may only be identifiable if the die size is small enough to generate sufficient detail in the wafer final test map. In this present study carried out on 300mm diameter silicon wafers, chip size was up to a few millimeters in size.

Coating defects are defects created during the lithography coating process, resulting in an uneven or irregular coating. Coating defects have a number of causes, including particulate, poor dispense timing, or a drip. After Develop Inspection (ADI) is used in semiconductor manufacturing as an inline control for the lithography process. [2] ADI controls, which are an in situ macro bright or dark field inspection with resolutions of 30 micrometers and above, allow the detection of both coating and incoming defects, where most coating defects caused during the lithography process can be removed through rework to improve yield. At the cost of resolution, most macro ADI systems can image each wafer without limiting tool speed.

Existing methods for macro ADI defect detection are known to have low sensitivity for certain coating defects. [3] This is due in part because nuisance defects from normal process variation can cause a similar amount of total color variation to real defects (e.g. incoming CMP (Chemical Mechanical Planarization), defects, ADI image stitching, edge topography, etc.). Thus, the recipe settings for color difference and affected area are insufficient for many processes. Despite progress in methods to distinguish critical and nuisance defects (including the use of global defect libraries) sensitivity and selectivity for color streaks and resist splatters remain a noted issue. Wafer test maps may indicate yield impact with a shape common for resist splatters (Fig. 1). These wafers would get reworked if the splatter was detected in situ during lithography. ADI may however have low sensitivity for finding such defects during the lithography process (Fig. 2). Detecting these splatters at the lithography step where rework is still available is of special interest because if three or more fields are affected by a splatter the wafer would get scrapped. Therefore there is an interest in improved detection methods for faint resist splatters and other global coating defects with large yield impacts. One method for improving ADI performance normalizes image color by tool intensity to create a more representative image set for model training and dispositioning [4]. For the defect model itself, commercial approaches also allow the model to be trained and adjusted to optimize sensitivity and selectivity.

Manuscript received 3 June 2022; revised 20 June 2022; accepted 22 June 2022. Date of publication 27 June 2022; date of current version 4 August 2022. *(Corresponding author: Matthew P. McLaughlin.)*

The authors are with Globalfoundries, Hopewell Junction, NY 12533 USA (e-mail: matthew.mclaughlin@globalfoundries.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSM.2022.3186607.

Digital Object Identifier 10.1109/TSM.2022.3186607

0894-6507 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.

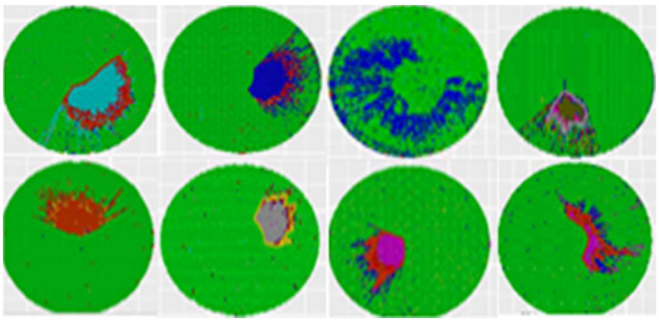


Fig. 1. Wafer test maps showing resist splatter yield impact when undetected at ADI: green color = pass, other colors are fails. The overall shape of these fails is common for splatters at ADI.

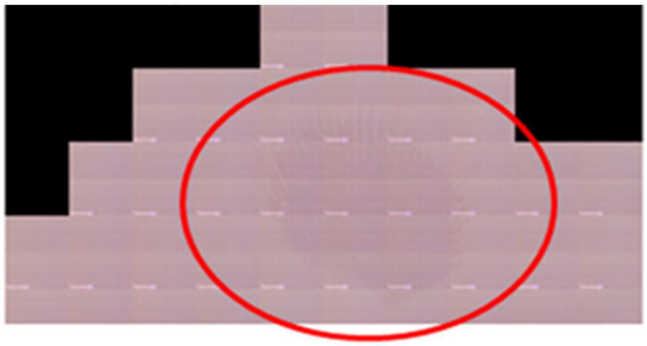


Fig. 2. Representative faint resist splatter at ADI circled in red, which correlates with yield loss. These defects are challenging to see and detect with ADI systems (See Fig. 3 for defect image enhancement).

But even with a commercial trained model, detection of faint color defects requires recipes to be sensitized to a remarkably high false negative rate (e.g. worst recipes had >50% of wafers dispositioned BAD where correct label should be GOOD), where improving the sensitivity and selectivity for the most subtle defects requires a system capable of distinguishing normal process variation from defects. The BAD label consists of multiple defect types lumped together under the BAD label for wafer dispositioning.

Herein we describe a Machine Learning (ML) method for detecting color and resist splatter defects in images for ADI dispositioning. Details provided on image enhancement by equalization are used to support ML model development and detection; the use of an explainable artificial intelligence (AI) approach to understand our model better; and a process flow ML training approach to systematically improve our models. Taken together, the complimentary ML system showed substantially higher sensitivity and selectivity for coating defects, which supported work to improve yield.

An initial study of this ML method for ADI was published elsewhere [5]. This current work expands on this initial project providing longer evaluation of the system's detection capability and line yield impact, describes the methods and infrastructure in more detail, and describes further exploratory work being conducted to further improve the sensitivity, selectivity, and yield impact of the system.

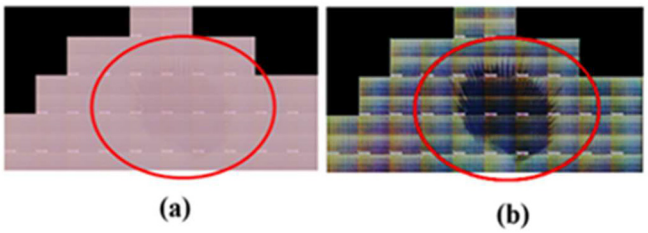


Fig. 3. Faint splatter defect from introduction (a) with defect circled in red. Defect more easily seen with image equalization used to enhance dispositioning (b).

II. Discussion and Methods

To improve sensitivity and selectivity of ADI for certain coating defects, we evaluated a complementary ML vison model using a cloud based AutoML service. This service became commercially available during the time of this study. We had prior success with the service from an earlier pre-release engagement that focused on defect classification of routine optical and SEM images of in-line semiconductor manufacturing defects. For this study, we were looking to extend the earlier AutoML model successes from the in-line defect work to ADI wafer maps images that would run in parallel to the current on-tool ADI module. From this work, a complementary automated disposition method was generated developed and tested for ADI that utilized an AutoML model that classified the ADI images into one of four labels: BAD, GOOD, NOIMAGE, and WRONGPATTERN. NOIMAGE refers to no ADI image being available for the respective inspection due to ADI tool hardware issue. WRONGPATTERN is a result of the wafer notch not being found in the expected location, and the ADI image thus does not match the reference image pattern direction. These labels are then assigned to thousands of representative images to teach the AutoML model how to predict for future unlabled images. During this work, methods in image enhancement, Explainable AI, and systematic ML model augmentation were useful supplementary tools used for model training and optimization of the two most challenging labels of BAD and GOOD.

To support ML model creation and evaluation of its predictions, a command line batch image processing program was used to automate the enhancement of images. In particular, a process was evaluated to equalize images, which provided a consistently superior visualization of defects for the human eye (Fig. 3). The enhanced images showed direct benefits, improving the accuracy and speed of dispositioning when validating the accuracy of model predictions data, and also provided indirect benefit by reducing mistakes from eye fatigue [6].

The equalized image also provided superior ML data preparation on defects that were hardest to see, limiting defect misclassification, which is known to undermine both the sensitivity and selectivity of models. The enhanced images also supported a rapid and confident dispositioning process for new test set data as it was being processed, which would then provide additional data to augment future training sets for the

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.

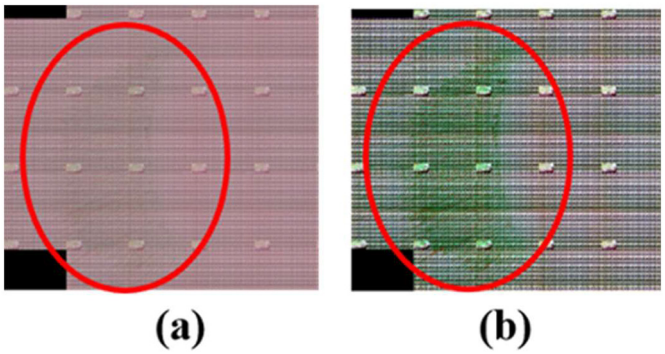


Fig. 4. Shows a faint splatter not detected with early ML models (a), which is more easily seen with the equalized images used to enhance dispositioning (b). The failure to detect this defect with ML contrasts with a similar defect in Fig. 3, which was detected with ML with very high confidence.

TABLE I
CONFUSION MATRIX FOR A PRELIMINARY ML MODEL WITH THE 4 LABELS USED. WHERE BAD = COATING DEFECT; GOOD = NO DEFECT; NI = NO IMAGE AVAILABLE; WP = WRONG PATTERN DIRECTION. THE ML PREDICTED LABEL IS SHOWN IN RED AND THE TRUE LABEL ASSIGNED BY AN ENGINEER IS SHOWN IN GREEN. THE CORRECT ASSIGNMENTS BY ML ARE FILLED WITH BLUE. ML FALSE POSITIVES AND FALSE NEGATIVES ARE FILLED WITH GREY, HIGHLIGHTING WEAK AREAS FOR THE MODEL

True/Predicted	BAD	GOOD	NOIMAGE	WP
BAD	78%	22%	-	-
GOOD	1%	99%	-	-
NI	-	-	100%	-
WP	-	-	-	100%

next model revision. As new production data was run against the current model, the resulting classified ADI images would be collected and batch equalized to provide a second, "more easily discernable" dataset for ground truthing that led to better data prep for future model training updates. For instance, the creation of the model described in this paper was achieved from six generations of ML model refinement, which required manual review of hundreds of thousands of images; using enhanced images allowed for a more rapid and accurate review of these images.

While refining each generation of the model, at times it was hard to understand why some defects were readily detected by the model and some were detected at a lower rate (see Fig. 4 and Table I). To aid in our understanding of some of the early ML model confusion, we explored the use of a beta version Explainable AI application to try and help identify what was driving the false fail predictions of the more critical defect types within the BAD label (where false fail denotes both false positives and negatives). The Explainable AI application provided saliency maps of the regions within the image that most correlated with what the model was using to make its prediction (Fig. 5). Using the saliency heat maps, we found that many of the false negative predictions were highlighting regions near the edge of the wafer map, even on the black border where no defects were located. By using the Explainable AI application, we were better able to understand how the model was visualizing the images, which then helped

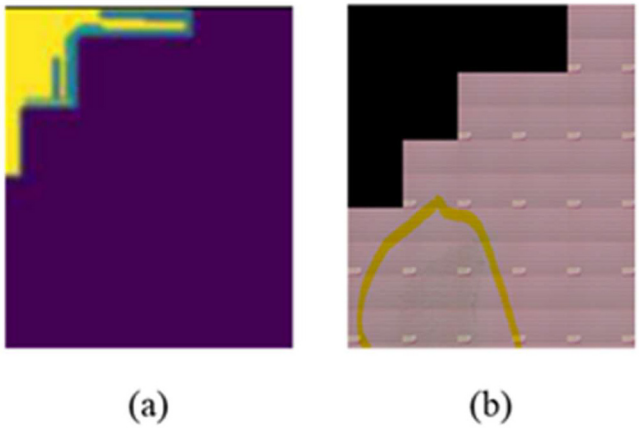


Fig. 5. (a) Explainable AI "heat map" showing the top left as having a defect (yellow) rather than in the middle left where the defect is actually located, circled in yellow (b). In other words, the preliminary models showed confusion on defect location even with a valid training set.

draw our attention to the variation in image dimension being produced by the on-tool ADI module and how the AutoML system managed those dimensions.

Investigating the ADI images revealed a wide range of image dimensions ranging up to a maximum of 4,256 pixels but the majority being under 2,240 pixels. Reviewing the AutoML documentation indicated the application had a limitation of 2240 pixels and for images greater than that the application would crop down to a maximum dimension of 2,240 pixels in either the X or Y dimension. This limitation was not encountered in the earlier in-line defect work as none of the optical or SEM images ever approached the 2240 pixel limit. This cropping was a problem for both images in the AutoML training set as well as images being sent to the model for a prediction. Without control of the image dimensions, we could have the defect removed from the image without our knowledge if the dimension were too large and/or the defect was close to the edge. To limit undesirable cropping of edge defects, which results in mislabeling and model confusion, we used a command line batch image processor to proportionally scale the dimension of all images to maximum of 2,240 pixels if the X or Y dimension exceeded that size. This one change improved the sensitivity and selectivity of the model, while enhancing the model's ability to detect edge defects, which were being cropped in our preliminary approach.

To further eliminate false negatives, from the perspective of the physical dispositioning of the wafers, additional filtering was implemented between the raw model label prediction and the label passed to the dispositioning algorithm. This filter consisted of a confidence threshold, wherein a predicted label of BAD would be treated as GOOD unless the confidence associated with the prediction exceeded a set threshold. This was implemented in such a way that the threshold could be set to different values for different attributes of the wafer being evaluated (Table II).

To have an archive and real time feed of model predictions to support model training and evaluation, a live feed of ADI images, ML model prediction and confidence was created

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.

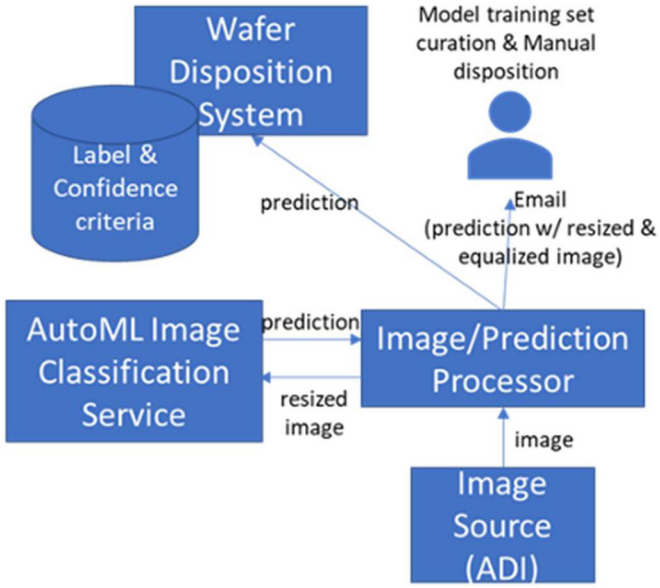


Fig. 6. System infrastructure flow diagram depicting how the image is delivered, processed, and used for prediction and future training.

TABLE II
MOCK ENTRIES IN A CONFIDENCE CONTROL TABLE III, WHERE FOR A BAD LABEL PREDICTION WOULD PASS DISPOSITIONING IF THE ML CONFIDENCE VALUE WAS BELOW THE THRESHOLD (E.G. A BAD LABEL WITH CONFIDENCE 0.3 FOR MASK LAYER AT LAYER1 WOULD PASS DISPOSITIONING). THE RULES OF THE CONFIDENCE TABLE DICTATED WHERE TWO CONFIDENCE VALUES WERE ASSIGNED TO THE SAME MASK LEVEL, THE HIGHER VALUE WOULD TAKE PRECEDENCE (E.G. LAYER1 W/ PROCESS PROCESS1 WOULD HAVE A CONFIDENCE THRESHOLD OF 0.6)

Label	Confidence Threshold	Mask Layer	Process
BAD	0.4	LAYER 1	*
BAD	0.4	LAYER 2	*
BAD	0.4	LAYER 3	*
BAD	0.8	LAYER 4	*
BAD	0.6	LAYER 1	PROCESS1

TABLE III
REPRESENTATIVE METRICS FOR PRELIMINARY ML MODEL SHOWN IN TABLE I. PRECISION INDICATES THE % OF CORRECT PREDICTIONS AMONG THE SUM OF PREDICTIONS FOR A GIVEN LABEL. RECALL INDICATES THE % OF CORRECT PREDICTIONS AMONG THE SUM OF ENGINEERING GROUND TRUTH DATA FOR A GIVEN LABEL

Total Images	7,787
Test Items	869
Precision	96.1%
Recall	93.3%

(Fig. 6). This feed was implemented via email and includes the resized image used for the prediction, the equalized equivalent of the image, and identifiable information about the wafer. This email served the purpose of providing a source of false positives (i.e., disposition of GOOD where correct label is BAD) and true positives images to be added to the next iteration of the model training set.

To ensure the continuous improvement of our model sensitivity and specificity for defects with each of the six generations of the model we compared each model's performance against thousands of images (Table III). If any of the ML metrics were lower than expected, the model could be redone using different training or test images.

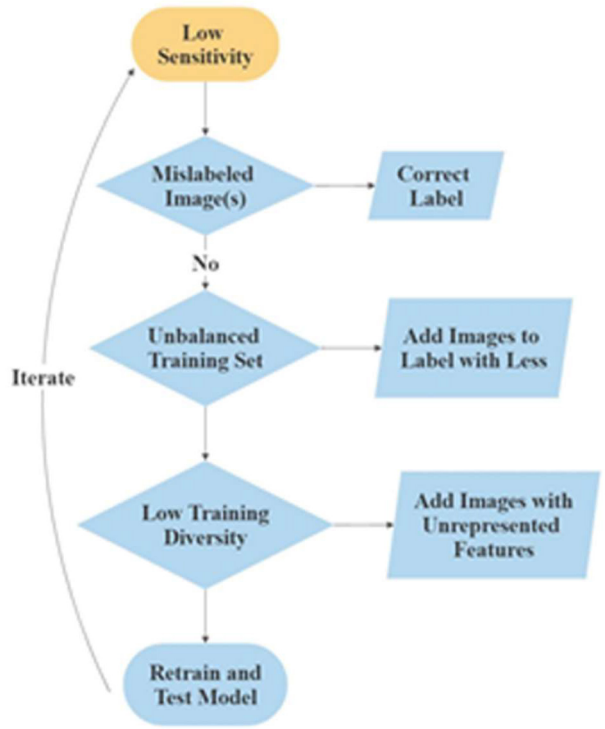


Fig. 7. Process flow diagram summarizing iterative training method to improve sensitivity for model prediction label features.

Through an iterative process of model refinement, a few practices were applied to optimize our AutoML model sensitivity and selectivity for defects, which are summarized in a process flow diagram (Fig. 7). For this application, sensitivity and selectivity optimizations focused on using performance to guide image augmentation, identification of inaccurate labels, and identifying the best labels to use. The best practices used have been well documented elsewhere, and so are only described briefly in the next two paragraphs [6].

To improve defect detection in each of the six generations of the AutoML model we sought to optimize sensitivity for the coating defects of interest. The practices to optimize ML model sensitivity for defects outlined in Fig. 7 focused on balancing the training set (i.e., using a similar quantity of images for each label) and identifying mislabeling in the training set. Since examples of defect types were limiting, the sensitivity of the ML model was optimized by using as many defect examples as could be identified over the evolution of the six generations of models. Even with all the training set updates we could still not identify enough defect examples for a balanced training set across all mask levels. For those mask levels showing low sensitivity, the training set was then augmented with manually edited images. These images were manually edited using a standard image processing software to make simple adjustments to Hue-Saturation-Brightness (HSB), or Red-Green-Blue (RGB) or simply drawing new pixels to an existing defect image and then saved as a "new" image to be added to the dataset. This was repeated multiple times to

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.

target a better balance across all available mask levels where training data was lacking. Additionally, if low sensitivity for a defect was identified, enhanced images of wafers labeled "GOOD" in the training set were reviewed for any mislabeling due to subtle defects which were originally missed. For example, Fig. 3 showed a subtle defect that could be mislabeled without careful review of enhanced images. Also of note, on two model training occasions, the randomly assigned AutoML training, validation, and test set images were determined to not represent a balanced sampling across all the various defect types that comprised the BAD label, This caused the models to converge with low sensitivity for the underrepresented defect types within the BAD label during training. To improve balancing the random assignment of training, validation, and test set images, an offline manual process using a list of image filenames that could be pre-sorted by defect type and mask level was implemented so that a quasi-random and more balanced sampling could be applied across each defect type sub-group and, where applicable, down to the mask level sub-group. These sub-group lists would then be combined into a single list that would then be used during the subsequent AutoML training to assign training, validation, and test set images.

We also sought to improve selectivity for the defects to reduce both false negatives and false positives. This required the model to have good sensitivity across different mask level images without any defects. Through the iterative process of model refinement, the method used to optimize sensitivity for defects was also applied to improve label sensitivity for wafers without defects (Fig. 7). Some more specific guiding principles were identified to maximize defect selectivity. For instance, selectivity was improved by including images with the GOOD label (i.e., without any defects) for each mask level. A clear pattern emerged, where unrepresented mask levels for the GOOD label received higher false negatives (i.e., incorrectly labeled BAD when correct label was GOOD) than mask levels with better representation. As a result, mask levels with more variation in the GOOD label had more false negatives. In cases where a represented mask level had persistently high false fails, additional images from the false fails were added to the training set to diversify the model. In instances where chronic false fails were observed for a mask level, the defect training sets with similar mask levels were inspected to identify if any wafers were mislabeled BAD (i.e., good wafers misclassified as having a defect).

The iterative training of a complementary AutoML model for coating defects was successful. Recognized ML best practices, [7] large training sets, and past learnings from automated use of ADI for defect detection all supported our success [8]–[10]. And although there are other examples of ML used to identify defects in the lithography area, [10]–[12] this work uniquely used image enhancement, Explainable AI, and an ADI defect library for defect model training.

III. Results and Conclusion

After six generations of model refinement, the detection method was compared against a commercial ADI system, showing over twice the sensitivity and three times the selectivity for global coating defects (Figs. 8 and 9). Each generation of the model was compared to the previous generation for several defect types, where sensitivity and selectivity were both evaluated (Fig. 10). Where an increase in the number of true

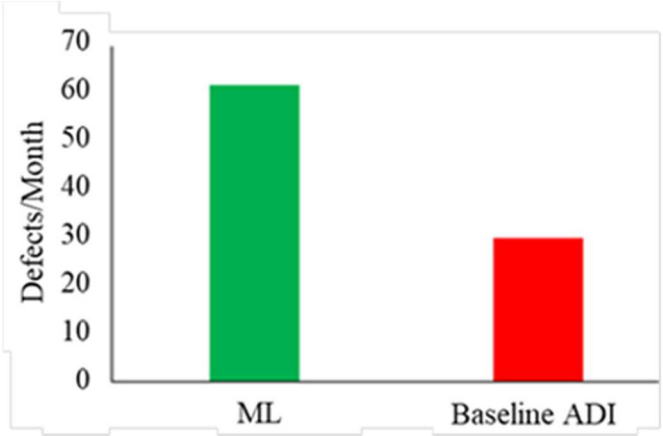


Fig. 8. Wafers detected comparing baseline and ML enhanced method in a controlled experiment. Shows the coating defects impacting 3 or more fields per month.

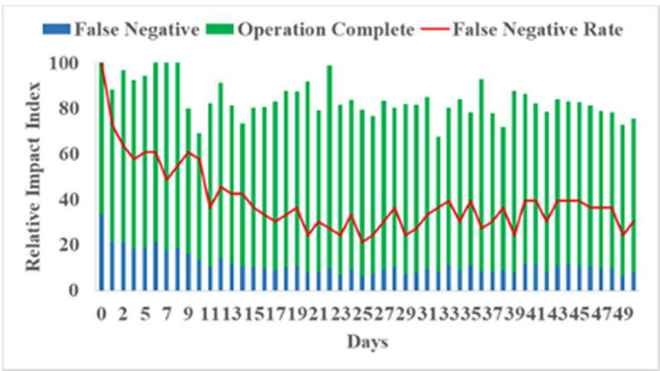


Fig. 9. Trend of ADI related false negatives expressed as a Relative Impact Index, shown in arbitrary relative units. An estimated 70% sustained reduction in false negatives (GOOD wafers mislabeled BAD by system) was achieved as the system defect selectivity improved.

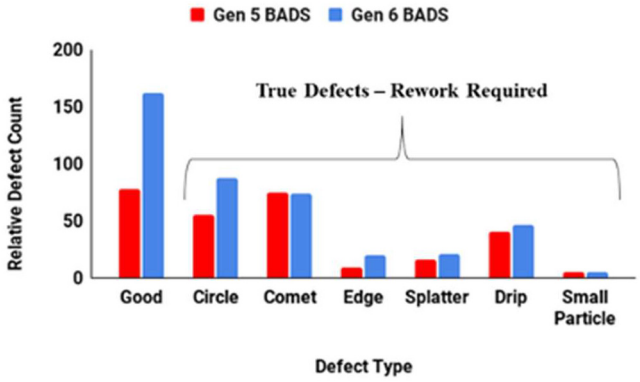


Fig. 10. Shows a comparison of the defect types that make up the BAD label assignments of ML Generation (Gen) 5 and 6 models compared for the same image set in an experiment. The higher defect sensitivity of Gen 6 is apparent from greater number of true defects detected but there were also more false negatives (i.e., Good wafers mislabeled BAD).

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.

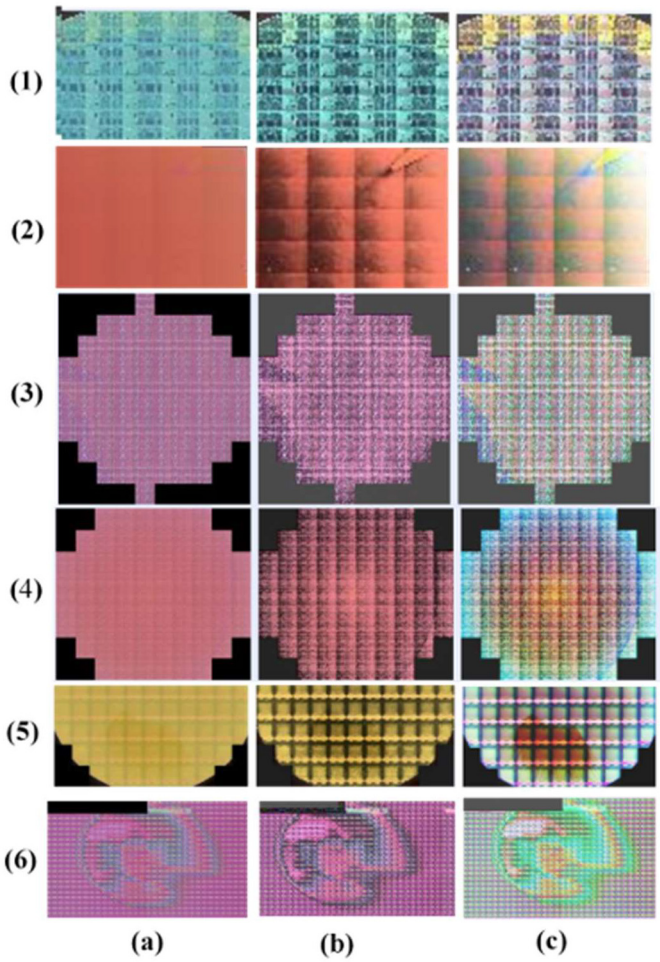


Fig. 11. BAD label representative images of coating defects shown as original image (a), HSB equalized (b), and RGB equalized (c). The defects shown include short shot (1), comet (2), edge (3), circle (4), splatter (5), and drip (6). Enhancing the image with HSB and RGB equalizations help to see the defect. Splatter often affects more than 3 fields, as seen in (5).

Fig. 12. Chart showing improvements in an inline Splatter Index over time. The Splatter Index is a description of the number of wafers dispositioned to have a splatter at ADI (arbitrary units). The time is shown as the project month.

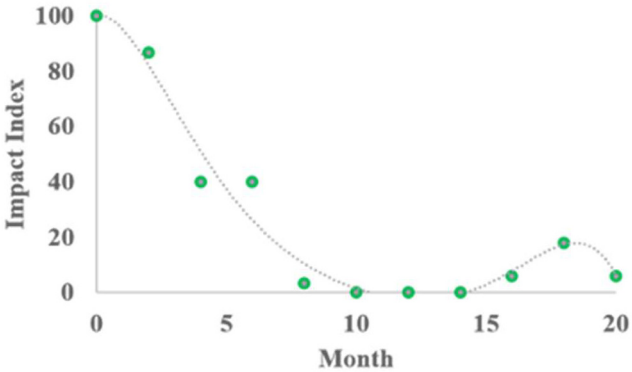


Fig. 13. Chart showing improvements in an Impact Index over time. The yield impact is shown versus months of project actions. The false positives, where a splatter was not detected by the ML, are utilized to improve the ML model and optimize other preventative actions.

defects labeled BAD indicated an improvement in the sensitivity of the model. The # of images mislabeled BAD (i.e. shown as 'Good' in Fig. 10) divided by the true defects correctly labeled BAD provides a measure of the selectivity of the model. A decrease ratio of mislabeled bad to true defects indicates an increase in model selectivity. As described in the methods, image enhancement was used to support better visualization of defects. Hue-Saturation-Brightness (HSB) and Red-Green-Blue (RGB) equalized images most consistently improved defect detection by manual inspection (Fig. 11).

During this study, increasingly sensitive ML models were used to detect defects for rework. The improving ML model supported and coincided with an increase in preventative maintenance and process improvements, which over time reduced the incidence of coating defects (Fig. 12). Consistent with the reduction in defects detected, a clear reduction in yield loss from coating defects was observed over the course of implementing ML to detect splatters inline. Splatter incidence was reduced approximately by 21 times (Fig. 13). The Impact Index is a description of the number of wafers determined to have yield loss from splatter incidents (arbitrary units). While the verification of yield impact of this work was confined to small die size wafers with full wafer final test maps, the authors of this study are confident that the effects, and thus the positive influence on yield, apply to any die size product. Depending on the failure, maintenance and process activities were developed. Including targeted routine actions, like the cleaning and replacement of failing nozzles, and process changes, including a switch to a walk-off rinse process.

IV. Future Work

In this work, it was found that certain mask levels and certain defects were more challenging to detect. For instance, if a defect existed only for an unusual or distinct mask level, the defect would be more likely to go undetected with a false positive (i.e., mislabeled GOOD by system when correct label BAD). The most likely source of this shortcoming is the existing method was based on a simple ML model with just two labels GOOD and BAD, where the absence of defects for certain mark levels would be expected to introduce confusion in the model. For commercial systems, this sensitivity shortcoming is partially addressed by using golden images. A golden image is a reference image, in this case, a defect free reference image of the whole wafer at that mask layer. A golden image is used for each mask level so much of

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.

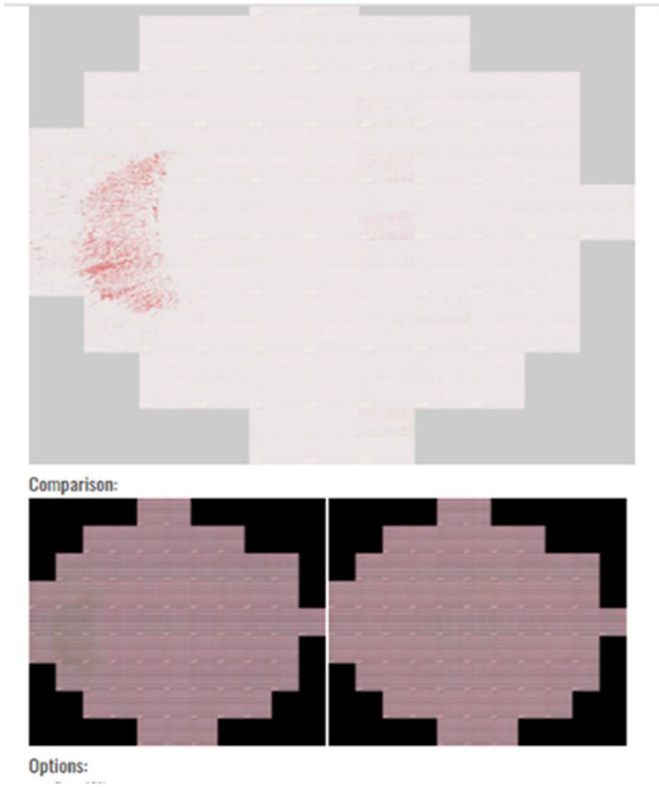


Fig. 14. Chart showing difference images between wafer without defects (bottom right) and with defects (bottom left). The difference image is shown on top with differences highlighted in red, better isolating the defect by removing the wafer features shown in gray in the background (i.e., the Golden Image).

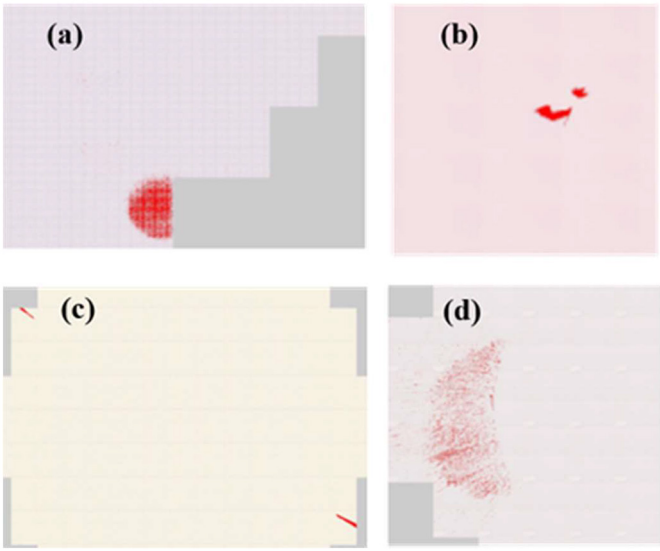


Fig. 15. Chart showing difference images between wafers without defects and with defects. The difference image is highlighted in red better highlighting the defect by removing the wafer features shown in gray in the background. Examples including (a) drip; (b) coating defect; (c) short shot; (d) splatter.

the mask level itself can be removed from the defect analysis. In this method, all mask levels were evaluated in a single model for efficiency in model development and cost, and so sensitivity was suppressed systematically for mask levels where no defects were in the training set. Outputs of the method did not have absolute quantitative references or classifications to control against, like commercial systems making it challenging to systematically exclude the most persistent nuisance defects (e.g., color confidence and number or size of different defects); largely nonvisual (even with image enhancement). For our future work, we have begun to explore the use of ML against difference images to create a more universal defect model (Fig. 14 and 15), which could be a more efficient system as it would only require new golden images for each mask level rather than continuous maintenance of a defect library.

Acknowledgment

The authors would like to thank Bruce Dyer and Rakhee Kumar for helpful discussions.

References

[1] R. Uzsoy, C.-Y. Lee, and L. A. Martin-Vega, "A review of production planning and scheduling models in the semiconductor industry part I: System characteristics performance evaluation and production planning," *IIE Trans.*, vol. 24, no. 4, pp. 47–60, 1992.

[2] T. L. Taylor, P. Shirley, D. Dixon, S. Yanagi, and E. Makimura, "Product and tool control using integrated auto macro defect inspection in the photolithography cluster," in *Proc. SPIE*, vol. 7638, Apr. 2010, pp. 763821–763829, doi: 10.1117/12.846571.

[3] M. Darwin, P. Kinikoglu, Y. Liu, K. Darwin, and J. Clerico, "Automatic classification of microlithography macrodefects using a knowledge-based system," in *Proc. SPIE Metrol. Inspection Process Control Microlithogr. XIX*, vol. 5752, May 2005, pp. 1024–1032, doi: 10.1117/12.599660.

[4] D. Dixon and L. Lee, "Method and system to compensate for lamp intensity differences in a photolithographic inspection," U.S. Patent 7 359 545 B2, Dec. 31, 2003.

[5] M. P. McLaughlin *et al.*, "Enhanced defect detection in after develop inspection with machine learning disposition," in *Proc. 32nd Annu. SEMI Adv. Semicond. Manuf. Conf. (ASMC)*, May 2021, pp. 1–5, doi: 10.1109/ASMC51741.2021.9435721.

[6] M.-J. Wang and C.-L. Huang, "Evaluating the eye fatigue problem in wafer inspection," *IEEE Trans. Semicond. Manuf.*, vol. 17, no. 3, pp. 444–447, Aug. 2004, doi: 10.1109/TSM.2004.831943.

[7] M. Zinkevich. "Rules of Machine Learning: Best Practices for ML Engineering." 2017. [Online]. Available: http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf

[8] V. C. Menon *et al.*, "Product and tool control using integrated automacro defect inspection in the photolithography cluster," in *Proc. SPIE*, vol. 6152, Mar. 2006, pp. 61521R1–61521R9, doi: 10.1117/12.656394.

[9] A. Berezin, D. Schaffer, and A. Villarreal, "Automating after develop inspection," in *Proc. IEEE/SEMI Adv. Semicond. Manuf. Conf. Workshop Theme-Innovative Approaches Growth Semicond. Ind.*, Cambridge, MA, USA, 1996, pp. 103–106, doi: 10.1109/ASMC.1996.557981.

[10] N. Rao, P. Kinney, and A. Gupta, "High throughput wafer defect monitor for integrated metrology applications in photolithography," in *Proc. SPIE Metrol. Inspect. Process Control Microlithogr. XXII*, vol. 6922, Mar. 2008, Art. no. 69223B, doi: 10.1117/12.772384.

[11] S. Purandare, J. Zhu, R. Zhou, G. Popescu, A. Schwing, and L. L. Goddard, "Optical inspection of nanoscale structures using a novel machine learning based synthetic image generation algorithm," *Opt. Exp.*, vol. 27, no. 13, pp. 17743–17760, Jun. 2019, doi: 10.1364/OE.27.017743.

[12] S. H. Lee *et al.*, "Machine learning approaches for nuisance filtering in inline defect inspection," in *Proc. 30th Annu. SEMI Adv. Semicond. Manuf. Conf. (ASMC)*, Saratoga Springs, NY, USA, 2019, pp. 1–4, doi: 10.1109/ASMC.2019.8791805.

Authorized licensed use limited to: BMS College of Engineering. Downloaded on August 21,2023 at 10:54:02 UTC from IEEE Xplore. Restrictions apply.