Wafer Defect Prediction with Statistical Machine Learning

by

Naomi Arnold

B.S. Industrial Engineering and Operations Research, UC Berkeley, 2009

Submitted to the Institute for Data, Systems, and Society and the MIT Sloan School of Management in partial fulfillment of the requirements for the degrees of

Master of Science in Engineering Systems

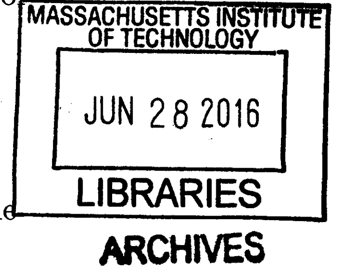
and

Master of Business Administration

in conjunction with the Leaders for Global Operations Program at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016


MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUN 28 2016
LIBRARIES
ARCHIVES

© Naomi Arnold, MMXVI. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author .... Signature redacted
Institute for Data, Systems, and Society and the MIT Sloan School of Management
May 9, 2016

Certified by .... Signature redacted
Roy Welsch, Thesis Supervisor
Professor of Statistics and Management Science, MIT Sloan School of Management

Certified by .... Signature redacted
Duane Boning, Thesis Supervisor
Professor of Electrical Engineering and Computer Science, MIT Department of EECS

Approved by .... Signature redacted
Maura Herson
Director, MBA Program, MIT Sloan School of Management

Approved by .... Signature redacted
John N. Tsitsiklis
Clarence J. Lebel Professor of Electrical Engineering, IDSS Graduate Officer

This page is intentionally left blank.

Wafer Defect Prediction with Statistical Machine Learning

by

Naomi Arnold

Submitted to the Institute for Data, Systems, and Society and the MIT Sloan School of Management on May 9, 2016, in partial fulfillment of the requirements for the degrees of
Master of Science in Engineering Systems
and
Master of Business Administration

Abstract

In the semiconductor industry where the technology continues to grow in complexity while also striving to achieve lower manufacturing costs, it is becoming increasingly important to drive cost savings by screening out defective die upstream. The primary goal of the project is to build a statistical prediction model to facilitate operational improvements across two global manufacturing locations. The scope of the project includes one high-volume product line, an off-line statistical model using historical production data, and experimentation with machine learning algorithms. The prediction model pilot demonstrates there exists a potential to improve the wafer sort process using random forest classifier on wafer and die-level datasets. Yet more development is needed to conclude final memory test defect die-level predictions are possible. Key findings include the importance of model computational performance in big data problems, necessity of a living model that stays accurate over time to meet operational needs, and an evaluation methodology based on business requirements. This project provides a case study for a high-level strategy of assessing big data and advanced analytics applications to improve semiconductor manufacturing.

Thesis Supervisor: Roy Welsch
Title: Professor of Statistics and Management Science, MIT Sloan School of Management

Thesis Supervisor: Duane Boning
Title: Professor of Electrical Engineering and Computer Science, MIT Department of EECS

The author wishes to acknowledge the Leaders for Global Operations Program for its support of this work.

Acknowledgments

This thesis would not have been possible without the help and support of many people. First I would like to thank my SanDisk supervisor, Yew Wee Cheong, for his unwavering dedication and championing of this project. A big thank you to Cindy You for her countless hours helping me - more than my "buddy" she was my confidante and friend. Thanks to the Quality Engineering team, Wenting Zhang and Xiaoqian Wang, for making me feel like part of the team.

I am grateful to SanDisk's LGO alumni for creating this amazing opportunity to work and live in Shanghai. I look up to Manish Bhatia, Kris Wilburn, and Weng-Hong Teh as leadership role models and they have been incredible mentors for me. A special thanks to the leaders across SanDisk that supported and contributed to this project. Thank you KL Bock and Gursharan Singh for sponsoring this internship to be in Shanghai. Special thanks to Itzik Gilboa for his guidance and advocacy of my project across organizations.

I would like to extend a big thank you to the teams in Shanghai, Milpitas, and Yokkaichi. In Shanghai, thank you Lena Zhang, Feiyun Zhang, Li Deng, Robertito Piaduche, and Joseph Idquival for sharing their KGD and test program expertise with me. In the Milpitas office I appreciate all the time that Hung Nguyen, Jason Yabe, Junius Tjen, and Loc Tu spent teaching me and facilitating this project. The big data strategy and machine learning analysis would not have been possible without the guidance and collaboration of Janet George, Jin Huang, Amit Rustagi, and the rest of the Big Data team. Thank you Kikuchi-san and the Yokkaichi fab team for the valuable feedback and hospitality. Thank you SanDisk and the many amazing people not named here that I had the pleasure to meet and work with.

I am humbled by this opportunity to call Professors Roy Welsch and Duane Boning my advisors. Thank you for all the time providing me with modeling guidance, knowledgeable perspective, and visits to China and California for this project.

Lastly, I cannot thank enough my family, friends, MIT classmates, and LGO compadres that have given me everything from words and wisdom to technical assistance. Your encouragement means more to me than I can express in words. I feel grateful every day to have such wonderful people in my life.

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Contents

1 **Introduction** **11**

1.1 Project Motivation 11

1.2 Project Statement and Hypothesis 12

1.3 Thesis Overview 13

2 **Background** **15**

2.1 Flash Memory Industry 15

2.2 SanDisk 17

2.3 Semiconductor Manufacturing Overview 19

2.4 Background Summary 22

3 **Literature Review** **23**

3.1 Machine Learning Approaches to Semiconductor Manufacturing 23

3.2 Big Data Opportunities in Manufacturing 29

3.3 Literature Review Summary 32

4 **Current Process** **33**

4.1 Manufacturing Process and Test Flows 33

4.1.1 Wafer Fabrication 34

4.1.2 Cherry Pick Wafer Sort 34

4.1.3 Known Good Die Test 35

4.1.4 Assembly 35

4.1.5 Memory Test 36

4.2 Description of Data Sources . . . 36
4.2.1 Die Sort Data . . . 37
4.2.2 Known Good Die Data . . . 37
4.2.3 Memory Test Data . . . 38

5 Model Development and Results 39
5.1 Model Preparation . . . 39
5.1.1 Data Refining Approach . . . 40
5.1.2 Evaluation Metrics . . . 42
5.2 Wafer-level KGD prediction model . . . 46
5.3 Die-level KGD prediction model . . . 49
5.4 Die-level memory test prediction model . . . 52
5.5 Results Summary . . . 58

6 Recommendations and Implementation Plan 59
6.1 Modeling Recommendations . . . 59
6.2 Cherry Pick program . . . 62
6.3 Wafer fab . . . 66

7 Strategy for Applying Big Data and Advanced Analytics to Semiconductor Manufacturing 69
7.1 Framework to Evaluate Applicable Problem Sets . . . 70
7.2 Impact Assessment Methodology . . . 72

8 Conclusion 75
8.1 Results and Limitations of Model . . . 76
8.2 Recommendations for Next Steps for Model . . . 77
8.2.1 Enhance Model Data Sources and Logic . . . 77
8.2.2 Simulation of Operational Model over Time . . . 78
8.2.3 Optimize Related Sub-processes . . . 79

List of Figures

2-1 Flash memory development has outpaced Moore's Law [6] 16

2-2 Flash versus HDD price trends and PC SSD adoption [6] 16

2-3 Enterprise flash vs HDD cost projections 2012-2026 [32] 17

2-4 Companies with largest global market share in NAND flash market, 2015 [19] 18

2-5 Forecasted NAND flash memory market CAGR [5] 18

2-6 SanDisk commercial and retail storage solutions [22] 19

2-7 Example stacked wafer map [15] 21

2-8 High level semiconductor manufacturing process [7] 22

3-1 High level strategy for model-view-controller architecture [14] 24

3-2 Experimental approach for ML model development [14] 24

3-3 Aggregated data example per die [13] 25

3-4 Nested structure for die within wafer and lot [13] 25

3-5 Wafer map to defect bin approach [25] 26

3-6 Data size and model run time experimental study [25] 26

3-7 Comparison of algorithms and predicted defects [18] 29

3-8 Big data ecosystem proposal [17] 29

3-9 IBM architecture for big data analytics to cognitive computing [8] 31

3-10 Case study for IBM wafer pattern detection solution [8] 31

4-1 High level overview of relevant process steps 34

5-1 Summary of model approach phases 40

5-2 Example confusion matrix 42

5-3 Example receiver operating characteristic (ROC) in R 43
5-4 Example variable importance plot from random forest output 45
5-5 Number of important features vs prediction error in wafer-level random forest prediction model 46
5-6 Top thirty important features in the wafer-level random forest prediction model 47
5-7 ROC curves of three trials of increasing data set size 48
5-8 Effect of increased number of failure samples in training set 48
5-9 Comparison of the best algorithms by threshold vs incorrect predictions . . . 50
5-10 Comparison of the worst algorithms by threshold vs incorrect predictions . . 50
5-11 Top four KGD soft bin prediction accuracy 51
5-12 Die-level random forest KGD prediction results 52
5-13 MT model development assumptions and data insights approach 54
5-14 Comparison of data sets and important parameters using a CART algorithm 55
5-15 Impact of sampling training set pass records 55
5-16 Best test and validation set results from MT random forest model, variable importance plot and ROC curves 56
5-17 Best test and validation set results from MT random forest model, confusion matrices 57
5-18 MT predictions validation results using CART algorithm 58

6-1 Prediction results incorporated into decision-making process 63
6-2 Mockup of updated wafer map based on prediction model results 65

7-1 Decision framework for evaluating big data ecosystem opportunities 71
7-2 Framework to map problem sets to opportunities 73

Chapter 1

Introduction

This chapter covers a general overview and context to introduce this project and thesis content. Section 1.1 covers the semiconductor manufacturing process context and Section 1.2 reviews how this project addresses a main challenge within this process. Section 1.3 previews the project approach and structure of this thesis.

1.1 Project Motivation

In the semiconductor industry where the technology continues to grow in complexity and Moore's law simultaneously drives decreases in manufacturing costs, it is increasingly important to identify defective die earlier in the process to recognize significant cost savings [10]. SanDisk would like to predict defective wafer die before two in-line test steps occur at their Shanghai assembly facility (SDSS). At the SDSS facility, wafers arrive from the wafer fabrication facility (fab) and a "cherry pick" wafer sort step identifies wafers that will meet subsequent stratified performance criteria. Then a costly test process occurs, called known good die (KGD) testing. Die are binned by KGD test results into different quality tiers and assembled into final products. After assembly, the final memory test (MT) exposes additional defects.

The KGD and MT test processes require expensive test machines, long test times, and are a reactive quality control process. Thus, there exists an opportunity for wafers with a high proportion of defects to miss KGD testing and be automatically downgraded, saving

valuable test time. There also exists an opportunity to avoid value-added assembly work for die destined to fail memory test. Multiple die (two, four, eight, sixteen, etc.) are stacked on top of each other to create one final package. Die stacking multiplies the cost implications that each defective die has on the value of the final package.

1.2 Project Statement and Hypothesis

The primary goal of the project is to build a statistical prediction model to facilitate operational improvements across two global manufacturing locations. A prediction model could improve the accuracy of wafer and die sorting, resulting in decreased assembly costs. It could also enable coordination with the wafer fab to implement root cause fixes to further drive cost savings. The main objective is to experiment with statistical prediction models to identify key values and correlation between die sort quality data (originating at the wafer fab) and test result data (KGD and MT that originate at SDSS). A second objective is to provide recommendations as to how a prediction model can be implemented in production to increase test yields and decrease operational costs at SDSS and at the wafer fab. Lastly, a third goal is to use this model as a case study to develop a high-level strategy for applying big data and advanced analytics techniques to semiconductor manufacturing.

Predicting defective wafer die has several major challenges. The datasets generated at the wafer fab and assembly facilities are compromised of hundreds of input variables. Mass production volumes are on the scale of millions of die each day. Thus, more resources are required to analyze this "big data" problem set. Existing analytical methods are not accurate enough for predictions since prior quality analyses are usually based on sampling. Another challenge exists due to the nature of the problem spanning multiple manufacturing locations. Access to data and coordination between international facilities are longstanding issues. As a result, SDSS lacks insight into the meaning of the fab's die sort parameter data and the upcoming, continual changes in processing at the fab. Currently, the KGD test process applies program-specific tests in place of full insight into and control over the fab die sort test results.

The project's main hypotheses are to predict a binary outcome of the known good die

defect category ("KGD soft bin") and memory test defect category ("test block number" or "FH soft bin"). The KGD prediction uses the primary upstream fab die sort (DS) test result data (called SME1). The memory test prediction model uses SME1 data and two additional upstream data sets as inputs (fab low temperature test result data, SME2, and known good die test parameter results). Input data sets are explained in more detail in Chapter 5. The project aims to build two prediction models that can identify die defect categories to an acceptable accuracy level that would meet business requirements. If the hypothesis can be proven, then implementation of the prediction model would result in wafers skipping KGD testing and avoidance of memory test failures. A small yield improvement of the KGD or MT test would result in significant financial savings given the high production volumes.

1.3 Thesis Overview

The thesis is laid out in seven main chapters. Chapter 2 gives context on the industry, SanDisk as a company, and high-level overview of the semiconductor manufacturing process. In Chapter 3, relevant academic and industry publications are analyzed that provide examples of other machine learning methodologies in semiconductor manufacturing along with background in yield modeling and big data applications. Chapter 4 explains the manufacturing flow, starting from the fab to the final testing at the assembly facility, in order to provide context as to how a prediction model would fit into this process. Chapter 5 provides the details of the model development process and results of three prediction models. In Chapter 6, recommendations present how to incorporate the prediction model into the current process. Chapter 7 proposes a strategic analysis of how to assess big data and advanced analytics opportunities in a semiconductor manufacturing environment. Lastly, Chapter 8 summarizes the main findings and next steps to improve and expand the prediction model.

This thesis also covers analyses and assumptions about the prediction model formulations. Algorithm experimentation was included in modeling since we assume that different wafer technologies need different algorithms and tuning to optimize prediction accuracy, reviewed in Section 3.1 and tested in Section 5.3. Another hypothesis is that the prediction models require large sample size of defects. Given low defect rate, large data sets are analyzed

along with various sampling methods in Section 5.4. Another hypothesis is that the models need to be updated in an intelligent manner to remain accurate over time, due to constant wafer fabrication facility process changes. Thus, experimentation with date ranges in the training and testing data sets are included in Section 5.4. Another area of exploration was the computational performance of the hardware that could handle this size of data set. R Studio is utilized in an off-line model on external servers with in-memory processing. Many other platforms and tools exist; alternative options are detailed in Section 3.2.

Chapter 2

Background

Chapter 2 covers background on the flash memory industry, SanDisk company history, and a high level overview of semiconductor manufacturing. Section 2.1 provides a context on technology development according to Moore's law, the competitive landscape, and industry growth. Section 2.2 includes basic facts about SanDisk as a company, its line of products, and current manufacturing locations. Section 2.3 discusses the main process steps for any semiconductor manufacturer, how to calculate test yields, and failure modes.

2.1 Flash Memory Industry

SanDisk operates in the flash storage solutions market where NAND is the primary technology. NAND benefits from a small form factor, high performance write speed, and solid-state format which enables data retention without a power source (non-volatile). The flash memory market has been characterized by fast-paced technology developments that decrease the price per bit [21]. Increased density of memory die per wafer has resulted in the capacity of each chip each year to outpace Moore's Law (Figure 2-1).

NAND flash memory advantages have resulted in disruption of the storage market as seen in Figure 2-2 and 2-3. Flash memory has been adopted in a large range of devices, replacing hard disk drive (HDD) technology. Analysts project that "the rapidly lowering cost and higher performance of flash will result in a rapid adoption of flash to replace magnetic drives. Flash together with systems of intelligence will enable the integration of big-data

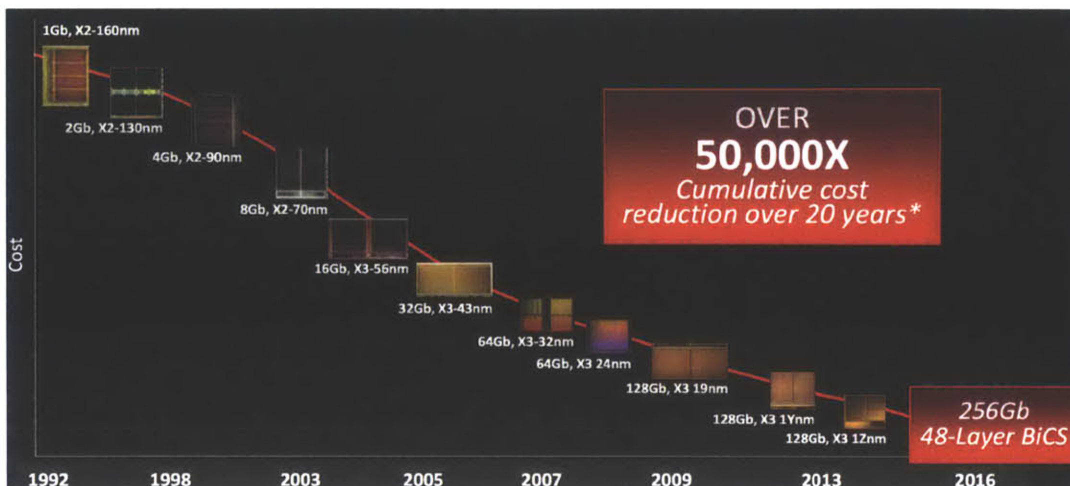


Figure 2-1: Flash memory development has outpaced Moore's Law [6]

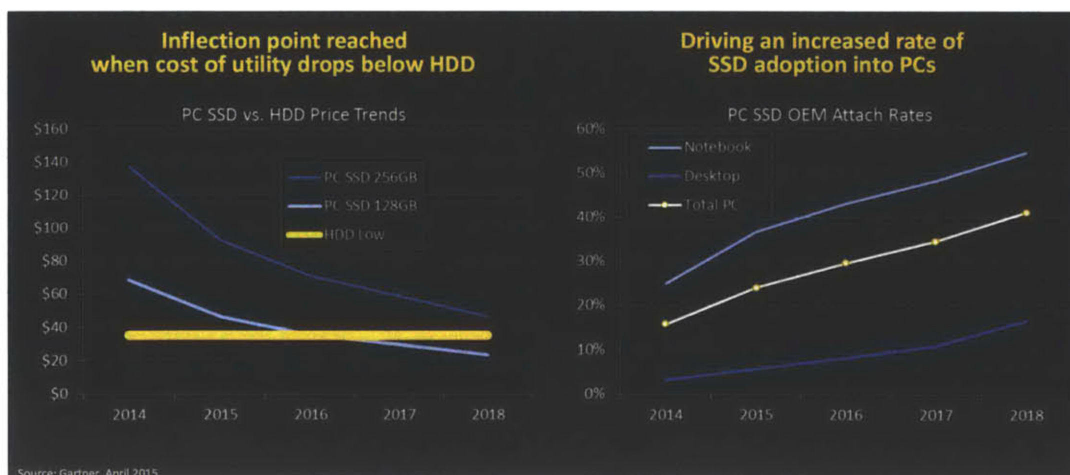
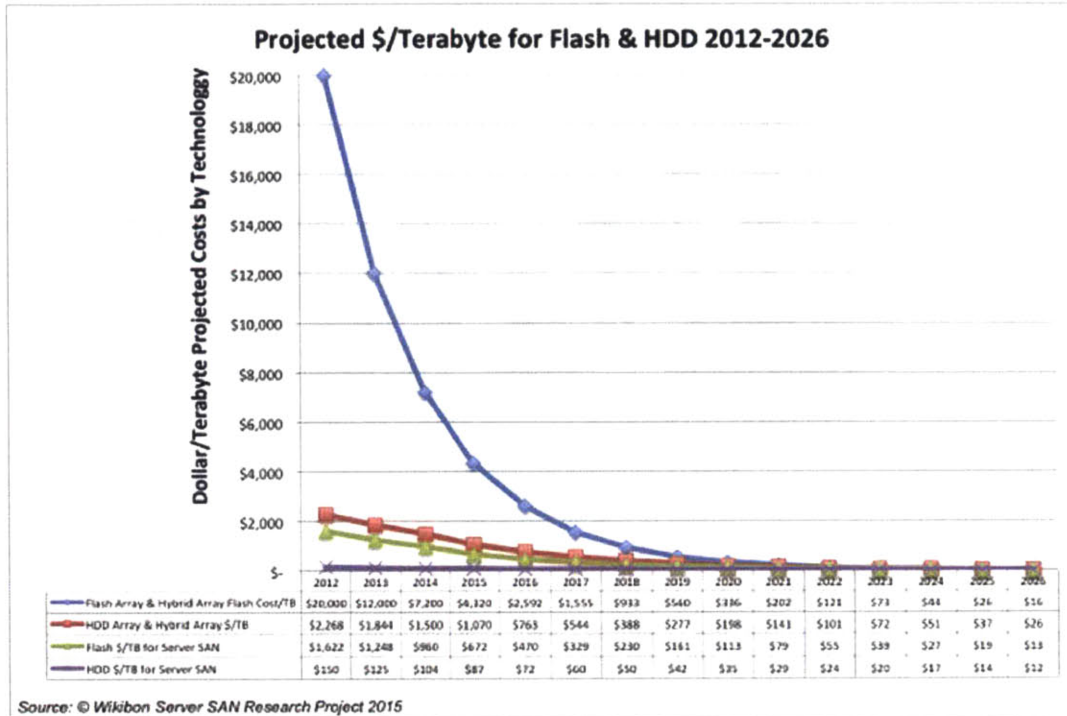


Figure 2-2: Flash versus HDD price trends and PC SSD adoption [6]



	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026
Flash Array & Hybrid Array Flash Cost/TB	$20,000	$12,000	$7,200	$4,320	$2,592	$1,555	$933	$540	$336	$202	$121	$73	$44	$26	$16
HDD Array & Hybrid Array $/TB	$2,268	$1,844	$1,500	$1,070	$763	$544	$388	$277	$198	$141	$101	$72	$51	$37	$26
Flash $/TB for Server SAN	$1,622	$1,248	$960	$672	$470	$329	$230	$161	$113	$79	$55	$39	$27	$19	$13
HDD $/TB for Server SAN	$150	$125	$104	$87	$72	$60	$50	$42	$35	$29	$24	$20	$17	$14	$12

Figure 2-3: Enterprise flash vs HDD cost projections 2012-2026 [32]

analysis into operational systems, and automate many decisions" [32].

The flash memory space is highly competitive. The main players include SanDisk, Samsung, Toshiba, Micron, Hynix, and Intel. Market share from Q3 2014 and 2015 are shown in Figure 2-4 [19].

In PricewaterhouseCooper's mobile technologies index, the compound annual growth rate for NAND flash memory was estimated to be 35 percent between 2011-2015, measured in megabytes per dollar as seen in Figure 2-5. The slim form factor of the solid state drive (SSD), which uses NAND as the storage component, will soon be the standard for tablets and smartphones. With the drop in price, original equipment manufacturers (OEMs) are turning to SSDs for desktops, laptops, and servers for price-performance advantages [5].

2.2 SanDisk

SanDisk is the worlds largest pure play supplier of flash memory data storage products. SanDisk was founded in 1988 by Eli Harari, Sanjay Mehrotra and Jack Yuan. Its stock,

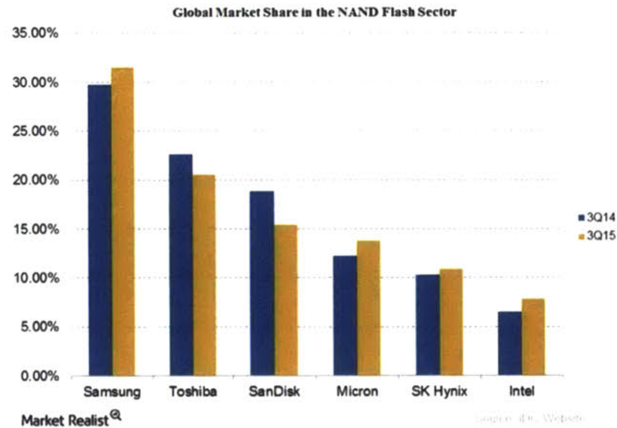


Figure 2-4: Companies with largest global market share in NAND flash market, 2015 [19]

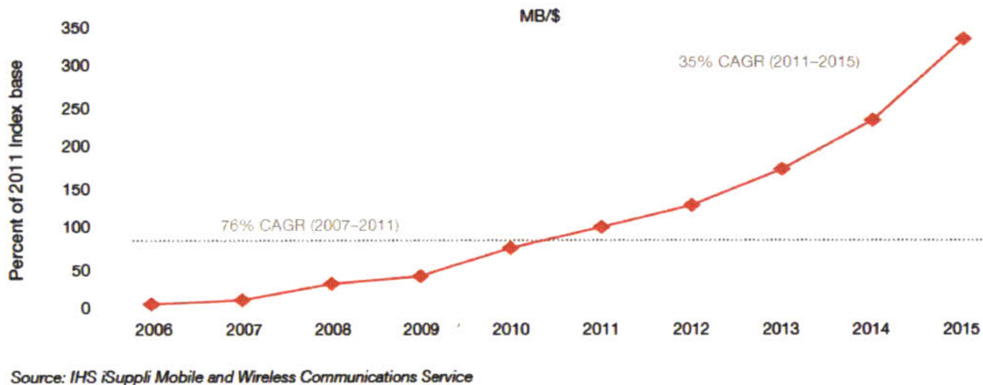


Figure 2-5: Forecasted NAND flash memory market CAGR [5]

SNDK, began publicly trading on NASDAQ in 1995. SanDisk is a US Fortune 500 company. In the past ten years, over one billion SanDisk memory cards were sold. SanDisk's mission is "to enrich people's lives through digital storage anytime, anywhere" and the company has about 8,600 regular employees. Products include flash storage solutions for enterprise data centers and client computing platforms (SSDs), removable and embedded flash products (mobile phones, cameras, automotive, connected home electronics, USB flash drives, DRAM, digital audio players, and SIM cards). Most products combine NAND flash memory with a controller and firmware, mostly designed in-house [21].

In 2014, SanDisk had revenues of 6.1B USD with commercial products (OEM's) accounting for two thirds and one third from retail products. Net cash in 2014 was 1.9B USD and

Figure 2-6: SanDisk commercial and retail storage solutions [22]

the company spent 0.9B USD on research and development investment that year. In October 2015, SanDisk announced that it will be acquired by Western Digital Corp. in a 19B USD deal [1].

SanDisk manufacturing facilities are located in China, Japan, and Malaysia. SanDisk Shanghai (SDSS) opened in 2006 and the facility covers assembly, test, packaging, and distribution of advanced flash memory products. SanDisk and Toshiba have a joint venture of the wafer fabrication facility in Yokkaichi, Japan. In 2002, SanDisk and Toshiba moved all of their NAND flash wafer production to the fabs in Yokkaichi. SanDisk has twenty global locations and is headquartered in Milpitas, California [23].

2.3 Semiconductor Manufacturing Overview

Semiconductor manufacturing is the process of taking a silicon wafer and adding layers that are patterned into integrated circuits (IC's). These are diced into individual die and packaged to be used in digital devices. This complex manufacturing process includes hundreds of steps.

Physical defects and variation that occur during manufacturing cause individual die and packages to fail to perform as desired. There are different categories of failures that affect semiconductor manufacturing yields. Parametric failures result from control deviations and

are associated with quality losses or functional failures. Area dependent failures occur on certain areas, such as foreign particles landing on the wafer. Random failures are uncorrelated and occur spontaneously. Generally, discrete failure probabilities are not Gaussian. They can be spatial in nature or based in binomial and Poisson statistics. Area dependent failures such as foreign particles cause different types of issues, such as distorted pattern layers, mechanical stress within the circuit if trapped between layers, or electrical shorts or opens. Non-local defects can also exist and are usually spatially correlated so they result in a clear spatial pattern; these may be caused by process variation [2].

The volume of die that perform as desired divided by the total number of die manufactured at that step is called the yield [7] and is usually reported as a percentage.

$$Yield = \frac{Good}{Total}\%$$

Yield can be calculated for each manufacturing test step and overall yield is the product of each of the steps.

$$Yield_{Total} = Yield_1 * Yield_2 * ...Yield_N$$

Assuming that a die will not perform as desired unless it is free of defects, the probability that a die successfully performs equals the probability that no defects exist on its area. Larger die area results in higher odds that it can include one or more defects. Wafers with large printed die have a lower die yield than small die wafers, even if they are developed by the same fabrication process and have the same defect density. Die yield losses from mis-processing can also escape detection in-line and during the parametric test. Some kinds of mis-processing impacts only a portion or die per wafer, for example edge loss due to less controlled film deposits near the edge of the wafer. A wafer map visualizes die yield by position on the wafer. As seen in Figure 2-8, an example stacked wafer map displays the average yield by die position [15].

Integrated circuit design results in a reference of electrical and physical characteristics of a complex electrical circuit device that uses semiconducting materials. A complete IC can contain millions of simple devices (resistors, transistors, diodes, capacitors) that work

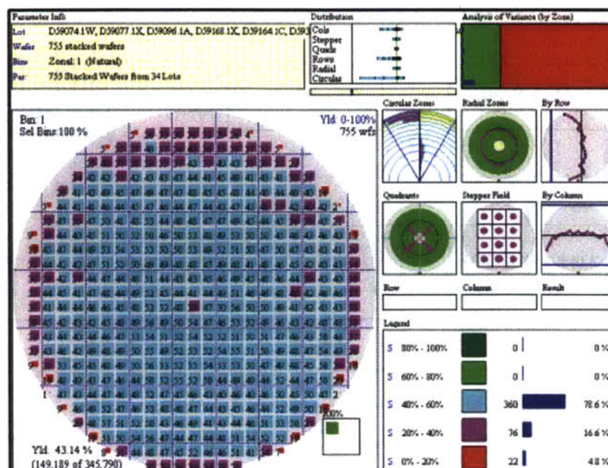


Figure 2-7: Example stacked wafer map [15]

together to perform functions such as memory or microprocessor. These IC designs are converted into layers that create electrical circuits when vertically integrated. The mask shop uses the design as an input to create a mask for each layer with geometric patterns. During wafer fabrication, many layers on each wafer are processed and each area of a single completed IC is called a die.

The wafers are made from silicon, which are grown from single crystals into silicon ingots. These ingots are sliced and polished, resulting in individual wafers. The wafer fab then performs hundreds of processing steps on the wafers. Wafers are oxidized by heating the wafers to a high temperature in an oxygen rich environment. Film deposition adds a layer of material on top of the wafer that is either conductive or non-conductive. Photolithography creates the pattern on the layer by applying photoresist, exposing the mask's circuit pattern, and developing the photoresist. Etch removes non-patterned surface layer or layers after photolithography. The photoresist is removed, leaving the patterned film on the wafer. In implantation, the wafer is doped with active ions to modify the properties of the wafer material as desired.

Each die on the wafer is probed, tested, and sorted so that bad dies are identified. Examples of testing include input and output voltages, current, signal timing, operational logic, and frequency of operation. "Bad" die fall into different failure categories ("binning") that assign a unique code. Hard bin refers to an overall pass or fail. Soft bin refers to a

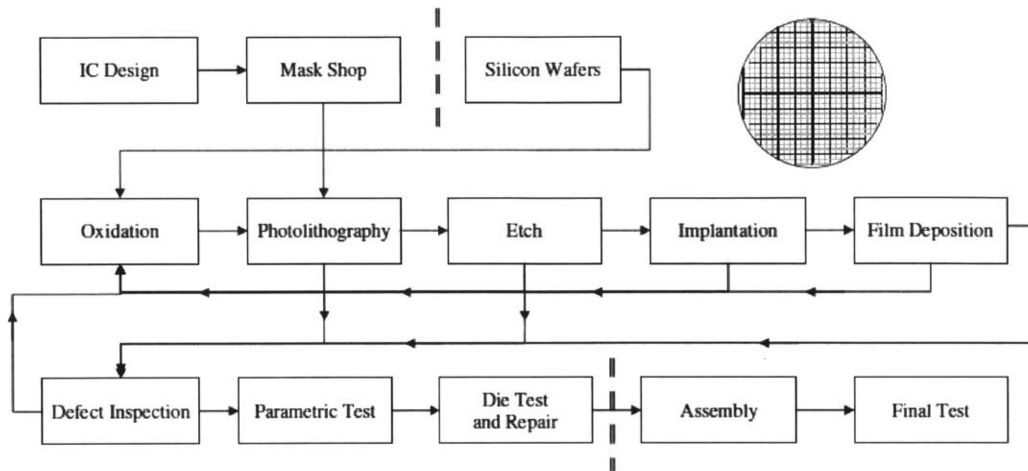


Figure 2-8: High level semiconductor manufacturing process [7]

categorization of failure, defined by test engineering. The yield is calculated at the end of this step.

The next main process is assembly. During assembly, wafers are transformed into individually packaged IC's. The wafer is cut (singulated) along scribe lines to separate each die. Each die, or stack of multiple die, is encapsulated in a package and metal connections are made. Packages can also be called chips or units. At the end assembly and packaging, final test occurs to ensure that the package passes electrical and environment requirements and final yield is calculated. Final test is the measure of the entire package's performance and meant to simulate extreme conditions of the real world [7].

2.4 Background Summary

SanDisk operates in a highly competitive, rapidly evolving flash memory industry. The flash memory manufacturing process is complex and capital intensive. SanDisk aims to gain competitive advantage in this space via technological innovation and streamlined operations. Chapter 3 will overview new areas of research in applying machine learning and big data to semiconductor manufacturing at other academic institutions and industry players. The new analytical capabilities provided by big data platforms, widely utilized in other technology and internet companies, can enhance current statistical quality control procedures at SanDisk.

Chapter 3

Literature Review

This chapter covers relevant industry and academic research in the areas of machine learning approaches in Section 3.1 and big data applications to semiconductor manufacturing in Section 3.2. Ideas from these references support and inspire the formulation of the prediction model methodologies discussed in Chapter 5. Section 3.3 summarizes the main takeaways from the literature review process.

3.1 Machine Learning Approaches to Semiconductor Manufacturing

A high level overview of the impact of applying machine learning to other manufacturing improvements is explored by Susto et al., covering virtual metrology, predictive maintenance, fault detection, run-to-run control, and modeling [26]. Challenges are outlined such as high dimensionality data, data fragmentation, time series input data, and multi process modeling. Many of these issues are encountered in our problem, as will be discussed in Chapter 5. Dimensionality reduction techniques are proposed by Susto et al. (principal components and correlation analysis, stepwise selection) to address the first challenge. Other ideas are proposed to help tackle the other challenges (data clustering and supervised aggregative feature extraction), yet validation and customization is found to still be needed on a case by case basis [26].

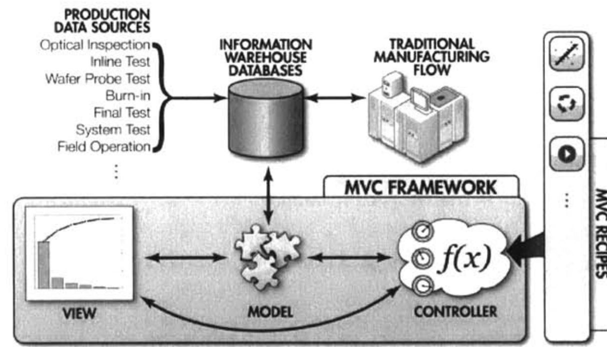


Figure 3-1: High level strategy for model-view-controller architecture [14]

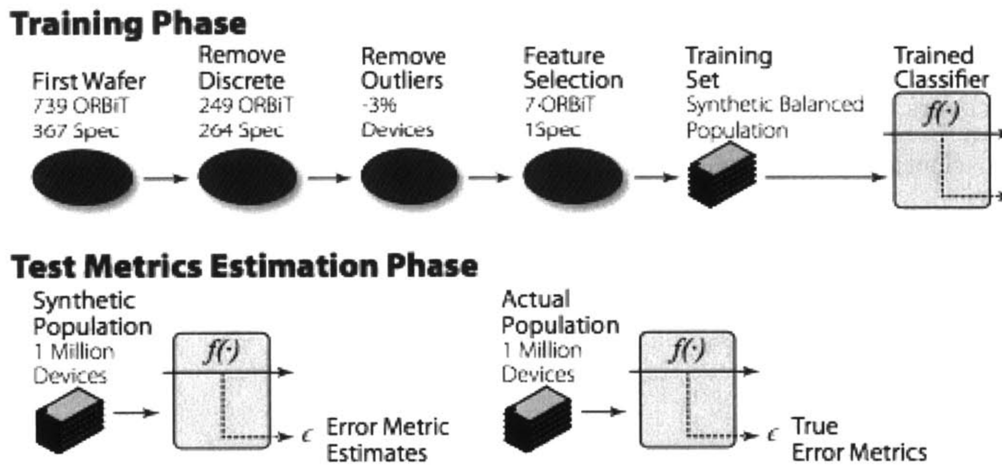


Fig. 7. Summary of experimental approach

Figure 3-2: Experimental approach for ML model development [14]

A TI/IBM case study by Kupp and Makris argues that studying data together from manufacturing and test flow is necessary to find process variability, since isolated statistical data analysis misses intra-process and test correlations [14]. The authors find that testing can account for up to 50 percent of the overall cost of an integrated circuit. Reducing test cost can be addressed by developing algorithms for post-production performance and spatial modeling of sparsely sampled wafer test results. The paper proposes a model-view-controller architecture for rapid iteration of complex machine learning methods to find optimal solutions with large datasets. The proposed framework and analysis methodologies are pictured in Figs. 3-1 and 3-2 [14].

LotID	WaferID	DieX	DieY	Defects on Layer 1	Defects on Layer 2	Bin	Fail = 1
1	2	1	9	0	2	1	0
1	2	2	7	1	0	1	0
1	2	3	12	0	1	1	0
1	2	4	3	1	0	1	0
1	2	4	15	0	1	1	0
1	2	5	6	0	1	1	0
1	2	5	7	1	0	1	0
1	2	5	8	0	1	1	0
1	2	5	10	0	1	42	1
1	2	5	11	1	0	1	0
1	2	5	15	1	0	1	0
1	2	6	15	1	0	1	0
1	2	7	3	1	0	1	0
1	2	7	11	0	1	1	0
1	2	7	15	1	0	1	0
1	2	8	8	1	0	24	1
1	2	8	16	1	0	1	0
1	2	9	1	1	3	7	1
1	2	10	1	0	1	77	1
1	2	11	1	0	2	1	0
1	2	11	17	1	1	1	0
1	2	13	16	1	1	1	0
1	2	18	8	1	1	1	0
1	2	18	9	1	1	7	1
1	2	18	10	1	1	1	0

Figure 3-3: Aggregated data example per die [13]

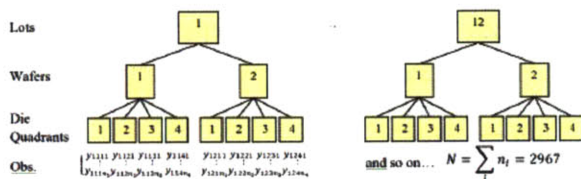


Figure 3-4: Nested structure for die within wafer and lot [13]

Krueger considers semiconductor yield modeling using generalized linear models, and provides valuable reference information for detailed approach methodology and data analysis [13]. The main strategy proposed to forecast yield is generalized linear models (GLMs) using defect metrology data. The research also integrates classification and regression trees (CART) with GLMs. The approach spans wafer-level and die-level analysis, and finds die-level predictions to be more accurate than wafer-level data sets, and that these performed better with larger sample sizes. The research shows that GLM models provide better predictions than the best historical model (Seeds Yield Model, an existing equation to predict performance based on defect count data). Krueger uses raw data structured in a similar manner as our data (described in Section 4.2), as pictured in Figure 3-3. The strategy of nesting die within wafers and lots as used by Krueger (Figure 3-4) is considered as a potential next step in our work in Chapter 8 [13].

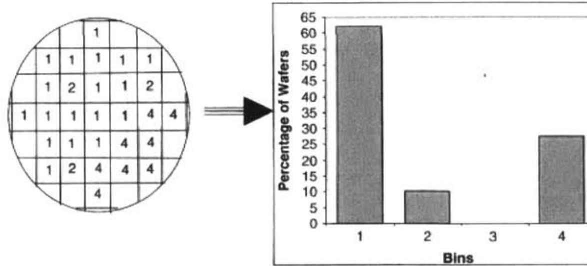


Figure 3-5: Wafer map to defect bin approach [25]

TABLE 1.1. Number of Points and Their Corresponding Run Time

Number of Points	Run Time (seconds)
325 (5%)	2,395.44
649 (10%)	12,453.28
974 (15%)	24,922.63
1,298 (20%)	51,112.13
1,623 (25%)	N.A.[a]
1,947 (30%)	N.A.
2,272 (35%)	N.A.
2,596 (40%)	N.A.

[a] N.A., not available.

Figure 3-6: Data size and model run time experimental study [25]

A methodology for defining the top defect bins to target is outlined by Soenjaya et al. [25]. The wafer map to defect bin approach (Figure 3-5) shows an example of a wafer that has 29 dies and four bins. Also pictured are the results from this experimental study's run time analysis, in Figure 3-6. The impact of increasing data size (2.5GB after data processing was complete) is seen to adversely affect run time. This analysis was performed using four months of STMicroelectronics wafer fabrication data on a Unix server Sunfire with algorithms implemented in Java [25].

Several publications describe successful experiments using a variety of machine learning algorithms. Support vector regression is successfully used by Lenz and Barak to improve virtual metrology [16]. They also note that approximately 50 percent of the time on the project was invested in data preparation. The authors state that simple and multiple linear

regression is not suitable for virtual metrology due to lack of accuracy and robustness. In their approach, predictive power is evaluated by the coefficient of determination (R) and accuracy is measured with mean square error (MSE), root-MSE (RMSE) and coefficient of variation. In terms of addressing outliers, their approach elects to not remove them, since it is assumed that important characteristics would be removed as well. This research had access to six months of process and metrology data, and found that only 1.6 percent of the original data set was suitable for analysis once wafers with missing data were removed. The authors also use expert input to identify the top six parameters and to decrease input parameter data set size before modeling. Results show R = .64 and CV(RMSE) = 1 percent, demonstrating the high predictive power of support vector regression in this problem [16].

Rosa and Vladimirov use another algorithm methodology, support vector machines (SVM), to improve quality control by early prediction of manufacturing outcomes [20]. In this approach, SVM with non-linear kernels on a per chip model tends to perform better, but not well enough to implement in production effectively. The authors did find that applying the SVM model to wafer-level classification results in the prediction accuracy for low-yield wafers to be as high as 81 percent. This paper highlights the potential advantages of wafer-level classification over die-level modeling [20].

Weiss, Dhurandhar, et al. use IBM data to discover that an ensemble method with boosted trees and linear regression to be the best performing machine learning algorithm [30]. The authors designate a proxy for microprocessor chip speed as the predicted outcome. 125 wafers (5 lots) are in the data set and data is sampled at 10 percent. Their approach also fills in missing values with a feature mean. The model data inputs are wafer control measurements, such as lithographic metrology and electrical measurements. Similar to Lenz and Barak, 90 percent of the data is missing since it was based on measurements. The analysis finds that independent time-ordered sets are advantageous over randomly sampled wafers or lots since results change over time and the population is not stationary. The authors recommend using a large test set that spans a long time frame, representing varying conditions and recent data for the training set. The authors show that the classical linear model usually performs worse than forests, but the authors hypothesize that in non-stationary environments, such as fab performance evolving over time, the linear method could win since

it tends to not over fit the data. The authors propose averaging the two methods of forests and linear regression. Similar to the results discussed in Chapter 5, this paper lists the technical difficulties of predicting time-varying populations and missing data. To address this challenge, the authors propose incremental updates to the models as new measurements are recorded. This methodology requires specialized algorithms to add new wafers and keep older wafers with additional information and knowledge of chip-making to create a new class of methods to predict chip performance over time [30].

Two other publications explore neural networks as potential algorithms. Wu, Zhang, et al. demonstrate a fuzzy neural network approach for die yield prediction can achieve better precision than the Poisson, negative binomial, and neural network models [33]. In Hsu and Chien, a hybrid data mining approach for pattern extraction from wafer bin maps explores clustering and neural network models to improve yields at a wafer fab in Taiwan [9]. These authors focus on spatial statistics to extract patterns associated with manufacturing defects but note that further research is needed to develop different methodologies to identify specific patterns [9].

A recent publication by Kang, Cho, et al. demonstrates that wafer map spatial factors have the ability to predict die-level failures in final test [12]. The model inputs are four derived variables pertaining to wafer map features. The model predicts two types of failures using random forest algorithm. The authors demonstrate that variables based on die position are relatively more important and that prediction performance may decrease over time. The authors propose that including data from assembly, wafer fab, and test will enrich the parameter data set. Chapter 8 recommends a similar next step for this analysis since spatial factors appear to be very important in final test prediction modeling [12].

Another publication by S. Park, C. Park, et al. finds a similar result about the importance of spatial factors for pattern recognition using feature based die-map clustering [18]. In this approach, different response variables are tested (A: column fail bit count (FBC), B: column and row FBC, C: detailed column and row FBC) by three different algorithms, pictured in Figure 3-7. Fail and Pass Classification Accuracy FCA, PCA) are the designated evaluation metrics. The authors state that big data analysis is essential for this study and propose feature extraction from FBC data [18].

PERFORMANCE OF 3 CLASSIFICATION MODELS

Model		Type A	Type B	Type C
Decision Tree	FCA	0.8225	0.7830	0.8354
	PCA	0.7972	0.8069	0.8352
Support Vector Machine	FCA	0.8635	0.6084	0.9489
	PCA	0.8795	0.6934	0.9619
Artificial Neural Network	FCA	0.4382	0.7478	0.9461
	PCA	0.7391	0.3785	0.2885

Figure 3-7: Comparison of algorithms and predicted defects [18]

3.2 Big Data Opportunities in Manufacturing

The following section provides references to recent publications exploring big data and its applications in manufacturing. Some publications are high level strategic architectures and others are research based, using fab data for empirical studies or simulation.

What is big data? Big data is the convergence of internet, business, and sensor data that necessitates a new generation of architectures for analysis. It is significantly larger in scale than traditional data sets and is usually measured in petabytes. Big data also usually contains high dimensionality (thousands or millions per element) and a large diversity of data (semi-structured or unstructured). The data is usually combined across multiple sources, flows at a rapid rate, and uses adaptive or machine learning-based analytics to handle the large data set size. As new hardware and software technologies arise to support the growth of big data, it is expected that opportunities for semiconductor players will be significant [10].

V. CONCEPTUAL BIG DATA ARCHITECTURE

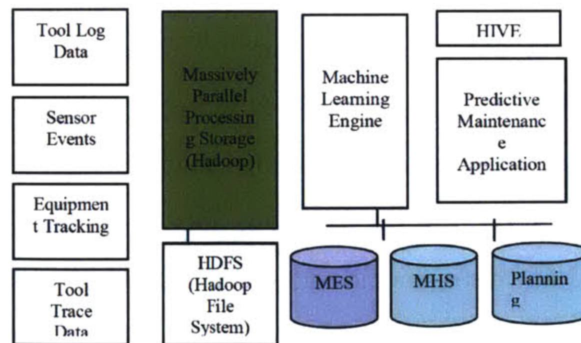


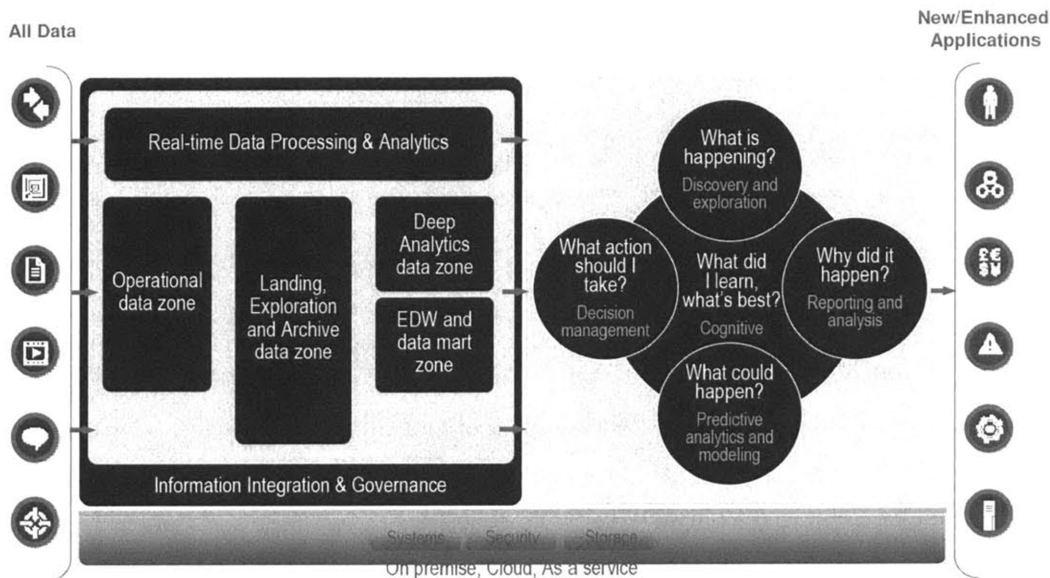
Figure 3-8: Big data ecosystem proposal [17]

Munirathinam and Ramadoss review various big data system architecture elements, such as Hadoop, NoSQL, and Random Forest, to describe how big data predictive analytics can be applied to proactive semiconductor equipment maintenance (Figure 3-8) [17]. The authors describe the potential to move the semiconductor industry from a reactive to a predictive state in the areas of virtual metrology, predictive maintenance, fault detection, run-to-run control, and modeling. They posit that these predictive states can prevent unplanned downtime, extend the useful life of semiconductor equipment, and improve product quality [17].

Wang and Alexander provide a comprehensive overview of various manufacturing fields and processes that can benefit from big data improvements to design and operations [29]. The authors summarize the four main characteristics of big data as high volume, high velocity (collected, processed, and visualized in real time), high variety (many types of information), and high veracity (accurate and comprehensive). They also prescribe the services required to support a big data environment: cloud infrastructure (storage, compute, virtual machine management), clusters, Hadoop related services/tools, analytics tools (logs, data mining, events), databases and servers (SQL, NoSQL), massively parallel processing databases, registries, indexing/search, and security provisions. The authors describe how big data can be applied across industries since it has applications in quality, time, costs, and mass-customization. For manufacturing engineering, the authors propose that the largest impact would be on detecting defects, boosting quality, and improving supply forecasting. Design and manufacturing opportunities are prevalent in other industries such as electricity, automotive, missile, integrated circuits, semiconductors, additive manufacturing, and medical devices [29].

An IBM report about big data and analytics for semiconductor manufacturing show IBM's microelectronics organization successfully integrates a big data platform and custom applications in their microelectronics organization based on IBM's analytics and manufacturing expertise [8]. IBM combines data from the fab (metrology, logistics, test, sensors, inspection) to predict yield analyses and identify top variables. IBM demonstrates the automation of data analysis and multivariate analysis of wafer test patterns to detect yield changes and sensor data in IBM fab to optimize assets and yield control (Figure 3-10) [8].

Chien and Chuang propose a framework for root cause detection of sub-batch processing

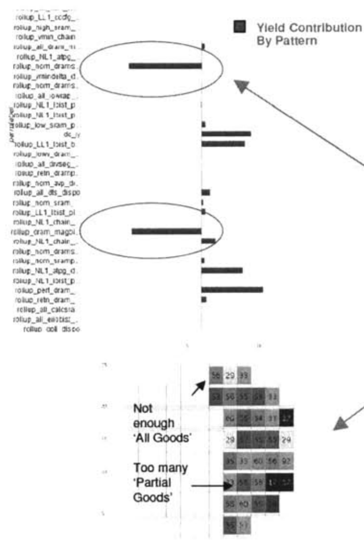


© 2014 IBM Corporation

Figure 3-9: IBM architecture for big data analytics to cognitive computing [8]

Use Case 1: Real-time multivariate analysis of wafer test patterns with Streams

Partial Least Squares (PLS) model compares actual yield to previous results
- analysis output highlights what has changed



Automated Streams solution:

- compares yield by test pattern to historical data
- identifies unusual yield behavior, based on multivariate model
- larger bars indicate larger deviation from historical yield
- has been used to immediately identify problems on leading edge of new production
- problem identified before the first wafer had completed testing
- new data added to existing model and kept in memory for fast and easy analysis

Benefits:

- 20% reduction in engineering labor
- first quality escape prevented - $650k in avoided warranty expense

© 2014 IBM Corporation

Figure 3-10: Case study for IBM wafer pattern detection solution [8]

system for semiconductor manufacturing big data analytics [4]. This publication finds that the primary factor to increase screening efficiency is to leverage the random forest algorithm in a sub-batch processing model that handles collinearity and dimensionality. This study validates the approach with a Taiwan fab empirical study and simulations. The authors recommend to implement these sub-batch processing systems in order to catch a small number of errors with high accuracy [4].

Tsuda, Inoue, et al. propose a similar strategy to apply big data to advanced semiconductor manufacturing using Panasonic in-line fab data [27]. Their approach incorporates a big data model with a fab-wide fault detection and classification system to stop equipment and lots automatically when a fault condition is detected. This data feed extracts equipment parameters and implements virtual metrology along with run-to-run functions that are aimed to to reduce process variation [27].

3.3 Literature Review Summary

Based on the literature review above, several data refining methodologies are adopted in the model development discussed in Chapter 5. Similar findings about modeling challenges are described in Section 3.1 are echoed by this thesis, such as computational limitations of big data sets [25], technical difficulties of predicting time-varying populations [30], data preparation accounting for a bulk of model development time [16], and data fragmentation [26]. In terms of data refinement, methodologies are adapted in Chapter 5 based on the analysis summarized in Section 3.1, such as outlier values remaining in the input parameter data sets [16], die-level granularity for modeling [13], selecting the top defect bins for analysis [25], and experimentation with various machine learning algorithms. Other methodologies are listed as next steps to enhance the model in Chapter 8, such as including spatial input factors, nesting the die within wafers, decreasing the input parameter data set size, exploring wafer-level granularity, using the feature mean to fill in missing data fields, and enriching the parameter data set with assembly, wafer processing, and more extensive test output data sets. Section 3.1 reviews other proposals, frameworks, applicable processes, and successful case studies from other companies to support the big data strategy discussed in Chapter 7.

Chapter 4

Current Process

Chapter 4 provides an overview of the current state of the manufacturing and test process along with a description of the data sources utilized in the prediction model. Section 4.1 covers the main steps of the manufacturing process within the wafer fab and assembly facilities that are related to this project's problem statement. Section 4.2 reviews the die sort, known good die, and memory test data source structures to provide an understanding of the data used for modeling in Chapter 5 and to highlight potential data challenges.

4.1 Manufacturing Process and Test Flows

The important steps in the current manufacturing process are highlighted in Figure 4-1. Every die is tested at the wafer fabrication facility through a standard test process that generates die sort parameters and bins. The wafers are shipped to SDSS and are sorted into different expected quality tiers based on the die sort data in a process called cherry pick. Each cherry picked wafer undergoes the KGD test, which vary by customer requirements and engineering analysis. During assembly, singulated die are attached to substrates and stacked to create various products (for example, 4 stack or 8 stack die). At the end of assembly, the final products are tested thoroughly before being released to customers or downstream processing (for example, solid state drives, SSDs).

There exist additional quality procedures (in-line, statistical process control, and engineering failure analysis) but these will not be addressed in this study.

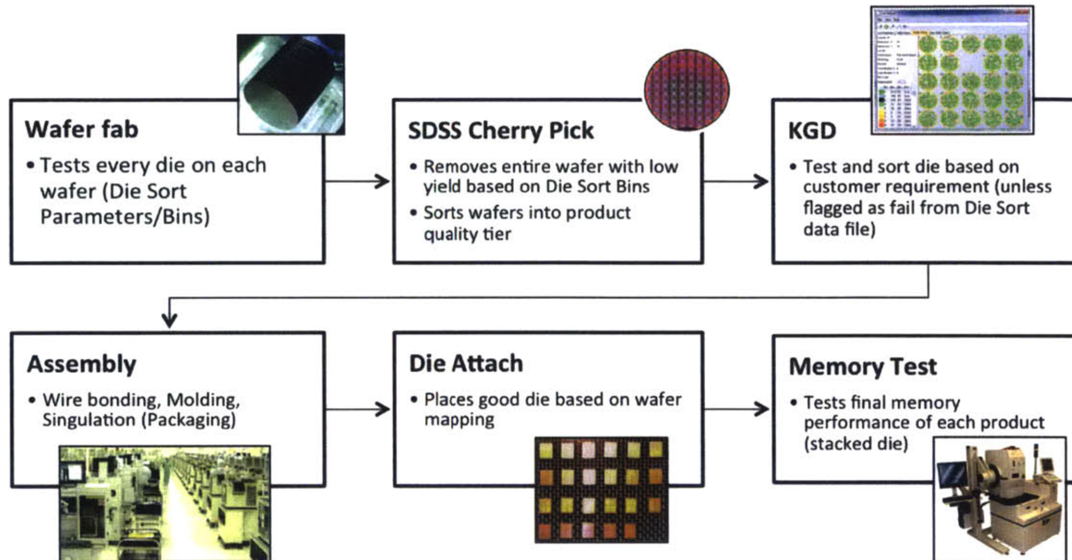


Figure 4-1: High level overview of relevant process steps

4.1.1 Wafer Fabrication

Two main data sets are generated at the wafer fabrication facility. The first is called SME1 (high temperature) and the second is called SME2 (low temperature). If a die fails SME1 then it is not tested in SME2. A die must pass both tests in order to be tested in KGD.

4.1.2 Cherry Pick Wafer Sort

The cherry pick process is controlled by product and test engineering experts. The purpose of cherry pick wafer sort is to identify wafers with high proportion of prime die that should be assembled into products that require the highest performance levels. These wafers undergo a more stringent KGD test program, and subsequent assembly and test processing differs as well. The cherry pick process controls how wafers are sorted by increasing or decreasing yield thresholds. These alterations also create levers to incorporate wafer availability and supply and demand in the market. New criteria are continuously reviewed and tested for all production lines and initially for new products during qualification stages. Criteria are statistically analyzed by experts to filter out die that exhibit known failure modes. Criteria are based on continuous die sort parameter output values. Each cherry pick rule set varies by product configuration so there is a large quantity of KGD test versions.

Incoming wafers are stored in a wafer bank on site at SDSS and volumes are planned weekly, based on demand. Wafers can also be sold direct to some customers. The physical cherry pick process runs non-stop and typically is completely automated except in some rare cases. The machines sort wafers from their incoming jar into three assembly lot cassettes that represent different quality levels. The machines run from lot files downloaded from networked servers that contain a predetermined wafer sort allocation.

4.1.3 Known Good Die Test

Upon arrival at SDSS the first test procedure, KGD, takes place before assembly processing begins in order to identify good and bad die according to customer specific requirements. Before KGD, wafers are prepared to mimic the customer placing the die on the substrate. This puts stress on the wafers so defective die can be discovered during KGD. There are hundreds of KGD tester machines that do simultaneous touch downs with two cassettes at a time. KGD testers generally run around the clock.

KGD failures are downgraded to lower performance products and KGD passes proceed to be assembled into prime products. A portion of die are skipped if they were already identified as a failure from the wafer fab die sort parameter testing. These may still be suitable for lower performance products, as a goal is to avoid wasted die wherever possible.

4.1.4 Assembly

The SDSS assembly process is broken into two main categories: front end (surface mount technology to wire bonding) and back end (molding to package saw). Inputs to assembly are the wafers that passed die sort and KGD and incoming materials, such as gold wire and substrates. First, a surface mount technology step occurs which includes solder paste mounting of passive components. Then, taping and grinding of the wafer is performed to protect it during wafer thinning (die preparation). After die preparation, singulation of each die with a diamond saw occurs and die are glued to the substrate. Wire bonding attaches gold wires from the wafer to the substrate circuit. Molding encapsulates each device with mold compound. The products are given ink marks or laser markings, such as the SanDisk

or customer logo. Lastly the package saw separates each final package.

4.1.5 Memory Test

At the end of processing when the products have been assembled in their final form, several tests are performed. The tests include a procedure to identify assembly-caused defects. A package preparation process occurs for certain products to stress each unit. The memory test verifies memory functionality and the factory high temperature (FH) is the most important memory test for most products. A low temperature test can also be applied for certain products. Some products are tested in both temperature levels and needs to pass both tests to be released to customers. A final test takes place which verifies the total performance of the product, called the system test. These different test programs, temperatures, and package preparation processes depend on the product design and customer requirements.

There are hundreds of memory test machines and each machine tests one lot in order to avoid mixing up the lots. The key goal of the memory test is to reduce test time since the memory test machines are the main bottleneck in the manufacturing flow, take up significant space, and are very expensive. Die can end up in nine different hard bins after memory test and some may be be re-tested due to test program issues or setting errors. Two types of testers exist (older and newer) and only data from the older testers are included in this study; products are tested randomly between tester types, so the fact that only data from the older tester version is included in the model is ignored.

4.2 Description of Data Sources

Test data is stored in one data repository system but requires manual extraction and merging. The following section describes the data sources involved in building the model. Die sort test data originates in the machines of the wafer fabrication facility and is processed into smaller files per lot. These files are encrypted, compressed, and transferred to the server where the data repository is located. A customized process analyzes the files for the cherry pick process and loads the data into the repository. Similarly, KGD and memory test data originates in the test machines in SDSS. These raw files are processed, aggregated, transferred, and uploaded

to the same data repository.

4.2.1 Die Sort Data

Die sort parameter and die sort bin data exist in different tables within the structure of this data repository. They must be joined on lot, wafer, die x, die y coordinates as the unique key value. Die sort bin data contains a binary pass or fail attribute and one soft bin for each die. Parameter data contains each die sort parameter test name and its value. There are roughly 600+ die sort parameters in the data sets for this study; the majority are from SME1.

The following assumptions are made about the die sort data and model formulation based on expert input from SDSS and the wafer fabrication process engineers. The wafers originate at multiple fabrication facilities (within the same site). Fabs are not separated by location since the assumption at the time was that processes and materials mix between locations so often that the resulting wafers are interchangeable.

The model ignores wafer spatial location and radial distance and just focuses on the input die sort parameter data alone. As discussed later in Chapter 8, calculating spatial indicators could be a promising next step. At this time, there is no access to a comprehensive reference list of each parameter and its meaning, obsolescence, or thresholds. Thus, logic rules about parameters are not implemented, such as dependencies between parameters, excluding irrelevant parameters, or grouping parameters. Lastly, the die are treated as independent units. A next step could be to create a nested structure where die are nested within wafers and wafers nested within lots in order to represent these relationships, as described in the data preparation methodology of Krueger [13].

4.2.2 Known Good Die Data

KGD data results are taken from a separate, live test system and the data is aggregated and sent to the data repository. During the KGD test program execution, there are hundreds of potential soft bins. Each die can land in only one soft bin and each test item occurs in chronological order. This model assumes that any untested bins would pass since once a die

fails a test item it does not continue the rest of the tests. This optimizes test time and is based on the assumption that test items are ordered by importance. As discussed in Chapter 8, there is a potential to explore this assumption to analyze soft bin correlations. KGD data also includes parameters values similar to die sort parameter data. For this study, around 300+ KGD parameters are included.

4.2.3 Memory Test Data

Memory test data is stored in several tables in the data repository. Die-level memory test results exist with an equivalent failure category to KGD soft bins, called FH soft bin or MT test block number. For consistency, it will be referred to as FH soft bin. Similar to KGD soft bins, each die has a binary result. The result is pass/fail and what single FH soft bin it landed in. As discussed in Chapter 8, the potential to reference continuous values would enhance the model but is not available at this time. For example, during memory test program execution, a bad block count accumulates for each die as it moves through each test item. There exists a cutoff value and if the bad block count exceeds this threshold, the die is identified as a fail and the die is not tested further in order to optimize test time. The test item when the fail occurred is identified as the FH soft bin. Thus, FH soft bin is a limited response variable since it contains little insight into the exact failure mode.

For this study, FH is the memory test step examined. For the memory test prediction model, specific memory test programs have been selected based on expert input. KGD programs are not filtered but assumed to be equivalent; this hypothesis could be explored further as discussed in Chapter 8.

Chapter 5

Model Development and Results

Chapter 5 summarizes the prediction models and analyzes the accuracy of the prediction results. Section 5.1 describes the methodologies developed while creating the prediction model, including data refining and evaluation metrics. Sections 5.2-5.4 cover the wafer-level KGD, die-level KGD, and die-level memory test prediction results. Section 5.5 summarizes the main results from the models.

Model development proceeds in three phases. First, a wafer-level proof of concept analysis predicts KGD failures based on SME1 data. Second, a die-level model predicts KGD failures based on SME1 data. Third, a die-level model is developed to predict MT fails based on SME1, SME2, and KGD parameter data. The general development strategy first focuses on predicting KGD results and then applying the lessons learned to the more complex MT model.

For each model, a specific technology, memory size (e.g., 64GB, 128GB), test program, and product are targeted. Collaboration with the relevant experts and data owners resulted in the model logic and data refining techniques described below.

5.1 Model Preparation

For all models, performance is evaluated with metrics of confusion matrices and receiver operating characteristics (ROC) curve. Recent historical production data is analyzed. Top soft bins and expert input determined which soft bins to select. Die-level granularity is

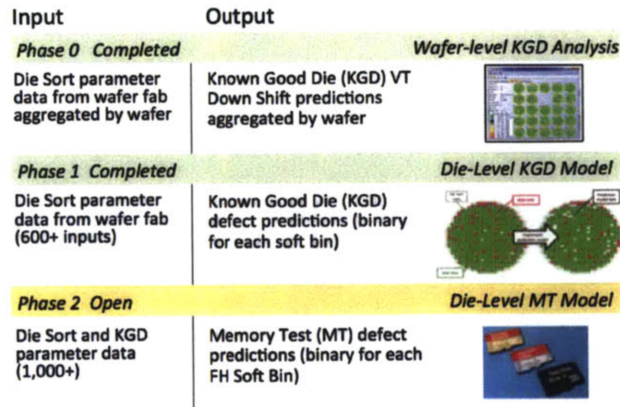


Figure 5-1: Summary of model approach phases

targeted since wafer-level analysis would not be useful for MT improvements. Plane-level analysis (there can be 1, 2, or 3 planes per die) would be a possible next step but at this time are assumed to not add as much manufacturing process improvement opportunities as identifying defective units at the die-level.

5.1.1 Data Refining Approach

The following describes the general methodology for data cleansing used to prepare the merged data sets of die sort, KGD, and MT test data. R Studio is used but other tools would be just as suitable, such as Python or a SQL database. Although rare, entire duplicate rows were removed. Any missing or null values are analyzed to determine if the entire field should be removed (majority of samples had null values) or just small numbers of samples that had data inconsistencies (very few samples had missing values for unknown reasons). Parameters with non-numeric values are excluded. Engineering input determined that outliers should remain and so the data set is left as is since data anomalies could be indicative of actual deviations. Samples per test date are plotted to ensure random, adequate samples over time (for the MT model). Engineering input determined that input parameters should all remain as is. Lastly, a sanity check of the number of lots (wafer lots, memory test lots), wafers, dies, defects, and remaining date ranges is performed before saving as a data frame in memory for processing.

The resulting data set is then filtered to focus on a specific device (technology), test

program, test step, SME1 date range, MT date range (for MT model), and part number (for KGD model) in order to compare samples that have the same test procedures and manufacturing flow. In the memory test data set, samples are removed that were KGD failures since they would have no MT response variable. In the MT model, samples are removed that engineering experts identified as not relevant to the analysis. Lastly, any other fields are removed (besides the response variable) that would not exist at the time a prediction would be made, such as hard bin.

A defect Pareto is next created to identify the top failure categories (soft bins). Engineering experts reviewed the top categories and the response variable, KGD or FH soft bin, is selected. For the MT model, FH soft bins are combined based on expert advice. All samples but the selected soft bin are updated to pass records. Only samples with the selected fail mode are designated by a binary "1" and all others (even samples that had a different fail mode) are designated by a binary "0". This model thus uses a binary classifier; Chapter 8 explores further methodologies to improve this categorization.

Next, the data set is randomly split into training and testing sets, split randomly two-thirds training and one-third testing. For a sub-analysis described in Section 5.4, the data set is split into training, testing, and validation sets. The validation set contained samples from a separate memory test date range to explore the impact of a separate time frame on the prediction results. The randomness of the split is verified by ensuring that there is roughly the same distribution between training and test sets of the percentage of failures. In cases where computational limits are anticipated to be an issue for the training set model formulation, pass records are randomly sampled (oversampling of the rare defect samples). This methodology will be described later in Section 5.4. Fewer samples in the training set results in a higher impact on the probability of overkill such that there are more extreme fluctuations of prediction accuracy. Experimentation with this data set shows that the number of pass records needs to be at least five times the number of failures to train a model that can reach optimal results. Thus, sampling the pass records is found to be a helpful methodology during initial development phase to speed up computation and still find a meaningful prediction level. Experimentation shows that at least several hundred fail samples are needed in the training set to obtain meaningful predictions.

Example test set	
real fails	3000
real pass	300000

```
> confusion.matrix(testing$MT_Fail,var_predict,.8)
     obs
pred      0     1
   0 299900   900
   1    100  2100
```

80% Probability

900	30.0%	underkill
100	0.03%	overkill
2100	70.0%	correct defects
	0.048	Overkill/Correct Defect Ratio

Figure 5-2: Example confusion matrix

5.1.2 Evaluation Metrics

The main evaluation metrics considered here are confusion matrices and receiver operating characteristic (ROC) to visualize model performance. Confusion matrices are calculated at probabilities ranging from 0.05, .1, .2, ... to .9 and manually analyzed to determine the optimal results.

In this model, there is a binary response variable. "0" indicates pass (negative for defect condition) and "1" indicates defect (positive for defect condition). In a confusion matrix generated in the R package *SDMTools* with *confusion.matrix* command [28], there are four values as shown in Figure 5-2. The top left is a predicted pass and actual pass (true negative, "specificity"). The top right is a predicted pass and actual fail (false negative, type II error, β, or "underkill"). The bottom left is a predicted fail and actual pass (false positive, type I error, α, or "overkill"). The bottom right is a predicted fail and actual fail (true positive, "sensitivity").

In Figure 5-2, the example confusion matrix shows the testing data set contains 303,000 samples, of which 3,000 are actual fails. After applying the trained prediction model to the test set, this confusion matrix is obtained. The prediction results in 299,900 true negatives, 900 underkill, 100 overkill, and 2,100 true negatives. The prediction identifies 2,100 defects correctly (70 percent) and misses 900 (30 percent). It also labels 100 as defects that are actual passes.

The confusion matrix in Figure 5-2 also introduces the concept of a threshold. The threshold is the third parameter input for the *SDMTools* package's *confusion.matrix* command after the testing set prediction model is complete [28]. Thresholds range from 0 to 1 and conceptually represent a likelihood. The particular results in Figure 5-2 correspond

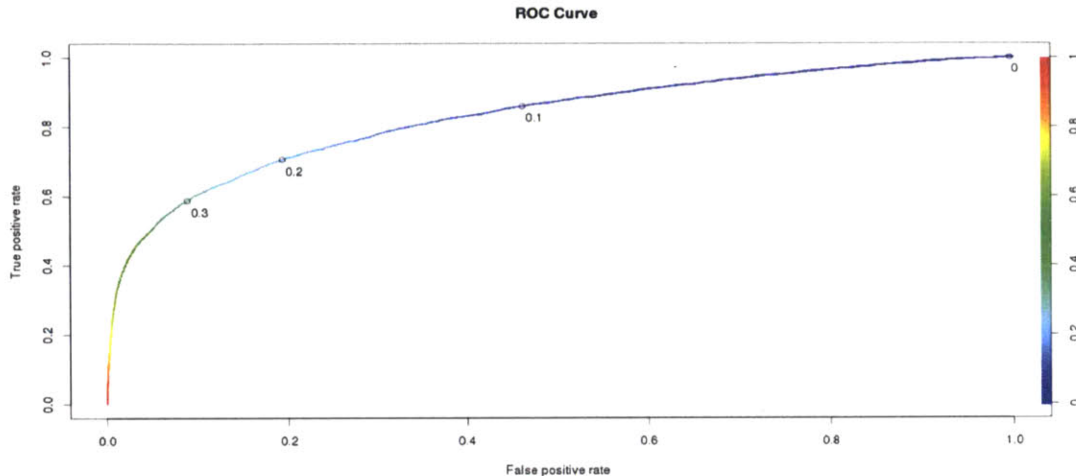


Figure 5-3: Example receiver operating characteristic (ROC) in R

to a threshold of 0.8. Threshold determines the sensitivity and specificity. Sensitivity is defined as the percentage of true positives (correct defect predictions). Sensitivity equals one minus Type II errors. Specificity is defined as the percentage of true negatives (correct pass predictions). Type I errors equal one minus the specificity [11]. Thus, alternative confusion matrices result when the threshold is changed. This threshold controls the expected incorrect predictions, which is a trade-off between false positives and false negatives. At higher threshold levels, false positives are reduced (defect prediction is more accurate) but false negatives increase (fewer defects are identified). At lower thresholds, false positives increase and false negatives decrease.

A receiver operating characteristic (ROC) curve visualizes the trade-off between the false positive rate (x-axis) and true positive rate (y-axis) at every threshold. The total performance of a classifier, summarized across all thresholds, is the area under the ROC curve (AUC). An ideal ROC curve would fit into the top left corner of the plot and maximize the AUC. This scenario represents a high true positive rate and low false positive rate. An ROC of a random guess would be a straight line from the bottom left to top right of the plot. ROC curves are a popular graphic since they simultaneously display both error types and consider all possible thresholds [11].

Figure 5-3 is an example ROC curve generated with the R package *ROCR* to compare model results [24]. A threshold of 0.8 lands in the orange portion of the curve in the

bottom left corner. At this threshold, false positives are minimized under one percent. As a trade-off, true positives remain around ten percent. Therefore, false negatives are around ninety percent. This example calls attention to another prediction modeling challenge of how to determine the appropriate threshold for future predictions. Namely, what threshold accurately represents the trade-off between false positives (lost revenue) and true positives (avoided cost)?

The rates of false false negatives, false positives, and true negatives alone do not provide the full picture. R squared is another metric that can be calculated with the prediction results, but does not provide enough insight into the prediction accuracy across thresholds to be a meaningful evaluation metric. Due to the small percentage of defects in the population, a metric of solely false positive rate will invariably be very small and misleading. In terms of incorporating business requirements, one hundred percent false negatives already exist in the manufacturing process as these are currently not caught in the status quo quality measures. On the other hand, false positives cannot be accepted since it represents wasted wafers or die. This waste represents lost revenue as this material would have been assembled into prime products if the prediction had not falsely flagged the unit as a defect. Given the high production cost already invested in these units by the time they arrive at assembly, there is a very low risk tolerance for any false positives.

This paper proposes a false positive ("overkill") to true negatives (correct defect predictions) ratio as an evaluation metric and recommendations are described in Section 6.1. This "overkill/correct defect ratio" compares the magnitude of false positives to true negatives. This ratio is a meaningful way to compare different prediction model results and determine the appropriate threshold value. For example, in Figure 5-2 an "overkill/correct defect ratio" of 0.048 is displayed (100/2100). If this ratio reaches one (2100 "overkill" and 2100 correct defect predictions), the prediction model would not make financial sense to implement since the number of false positives would reach the number of actual correct predictions. An appropriate ratio threshold can be set by the company based on the relative cost trade-offs between lost revenue of a wasted unit and cost avoidance savings of predicting defects.

For *random forest* output, a variable importance plot is generated with the R package *RandomForest* implementation using the *importance* command [3] and plotted with the

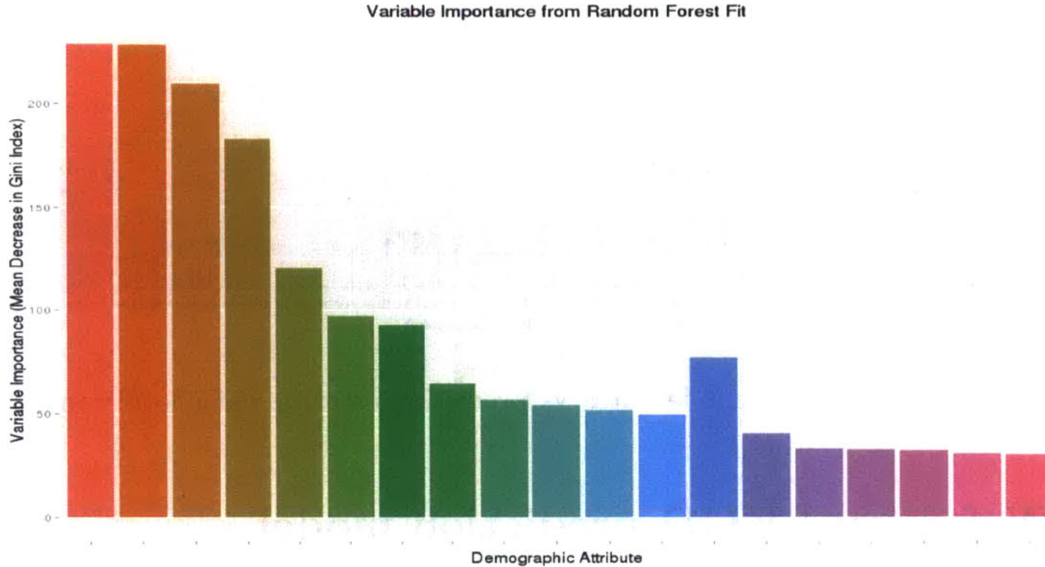


Figure 5-4: Example variable importance plot from random forest output

ggplot2 package [31]. The variable importance plot shows the input variables that the random forest algorithm identifies as being the most important in the prediction model based on a mean decrease in Gini index. In practice, classification error is not sufficient for tree-growing and two other measures are preferred, Gini index and cross-entropy. The Gini index, G, is defined as [11]:

$$G = \sum_{k=1}^{K} p_{mk}(1 - p_{mk})$$

p_{mk} represents the proportion of training observations in the mth region that are from the kth class. It is a measure of total variance across K classes. The Gini index becomes small when all of the p_{mk}s are near zero or one. Gini index is also called a measure of node purity. A small Gini index implies that a node contains mainly observations from one class. When building a classification tree, Gini index is usually used to evaluate the split quality since it is sensitive to node purity [11].

The mean decrease in Gini index is the y-axis on the variable importance plot. In Figure 5-4, each vertical column represents one input parameter from the training data set. The most "important" parameters are ordered from left to right and feature the highest variable importance score, expressed relative to the maximum [11].

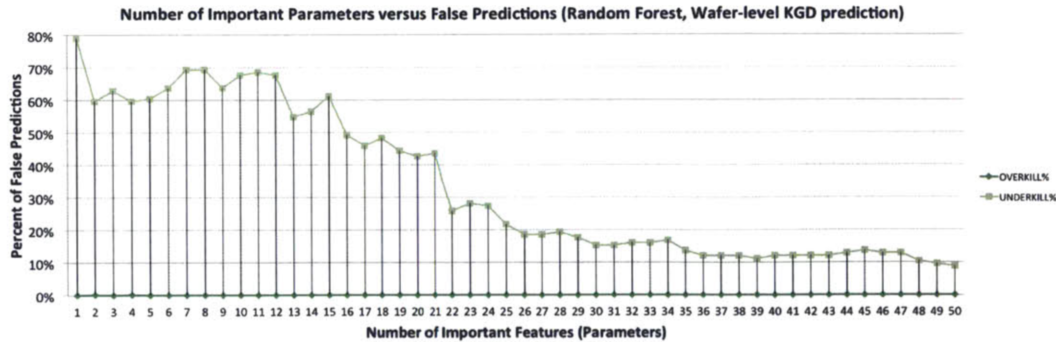


Figure 5-5: Number of important features vs prediction error in wafer-level random forest prediction model

5.2 Wafer-level KGD prediction model

Based on the data preparation described in the previous section, a wafer-level prediction model has been built in Python with random forest classifier. This proof of concept model proves that machine learning can successfully predict defects. This model also serves as an example for the die-level methodology, and several approach methodologies are based on subsequent analysis of the output. The model is based on a previous model built by another SanDisk team using python *sci-kit-learn* and *pandas* libraries. The data set is manually selected and merged from raw databases. This data set results in accurate predictions yet the data set is restricted in variability due to manual data extraction limitations; it only contains around 7,000 total samples in the training set and has a small proportion of failures. Random forest is identified as the algorithm with the least error after experimentation with multiple machine learning algorithms. The *sklearn.ensemble.RandomForestClassifier* default settings are kept as is except for three input parameters of the number of estimators, number of jobs, and minimum number of samples per leaf.

An appropriate response variable is selected in order to identify a particular defect mode of interest. Next, an analysis is performed to identify the number of important parameters. The first step of this analysis applies random forest to the entire SME1 input data set and ranks the top features in order. Then, fifty trials re-train the model with 1, 2, ... 50 of the top input parameters selected to comprise the training data. The results are shown in Figure 5-5. The x-axis displays the number of important input features present in the

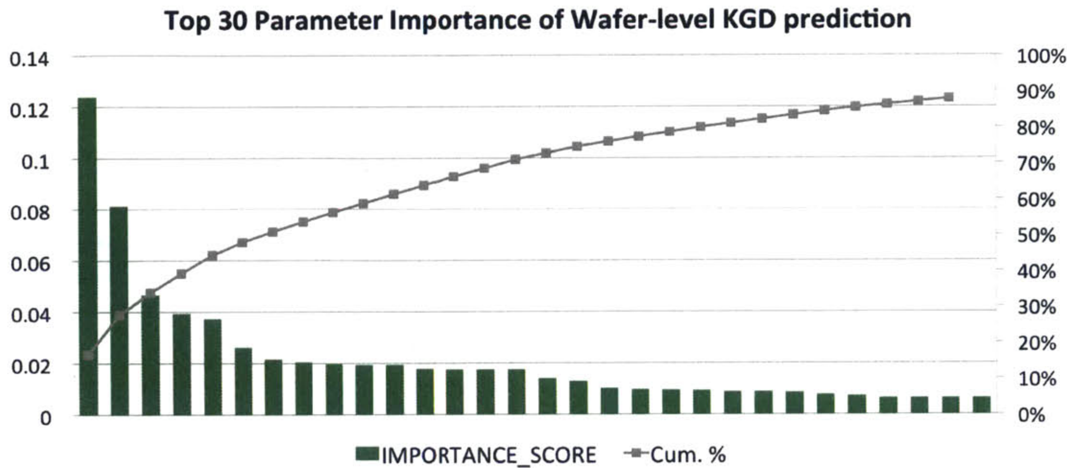


Figure 5-6: Top thirty important features in the wafer-level random forest prediction model

training data. The y-axis shows the percent of incorrect predictions. Incorrect predictions are separated into "overkill" (false positives) and "underkill" (false negatives). The percentage of false predictions are shown and calculated as overkill percent = overkill / population testing set size and underkill percent = underkill / actual defects in testing set. Overkill percent is less than 0.15 percent in all trials pictured. Around thirty features, the underkill percentage begins to plateau. This highlights an opportunity for feature selection in future model development. Narrowing down the input data set increases model computational speed without sacrificing accuracy.

Next, a variable importance score plot is generated as shown in Figure 5-6. Variable importance plots are described at the end of Section 5.1.2. Figure 5-6 displays the top thirty input variables from Figure 5-5 and their importance score as calculated by python's random forest classifier *feature importances* attribute. The parameter names have been removed from the plot. As shown by the cumulative percentage line, about 90 percent of the predictions can be explained by the top thirty input parameters.

It is hypothesized that the provided data set and prior random forest model needs to be expanded to include more samples to provide accurate predictions. An analysis is performed to show that with the addition of more data over time, predictions become more accurate. The three trials pictured in Figure 5-7 consist of an initial training set and adding one or two more data sets. ROC1 is the first data set. ROC2 is ROC1 plus more data that is sampled

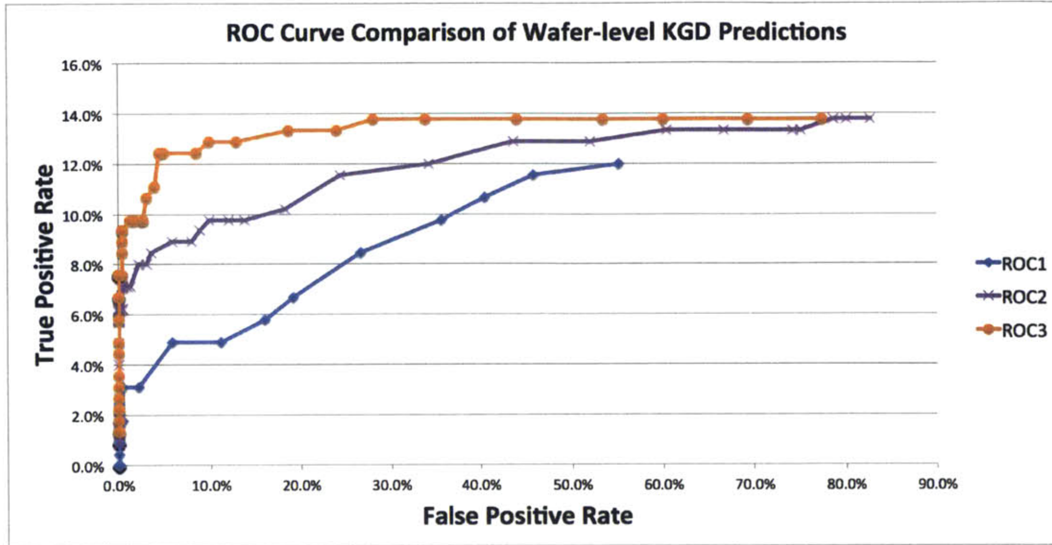


Figure 5-7: ROC curves of three trials of increasing data set size

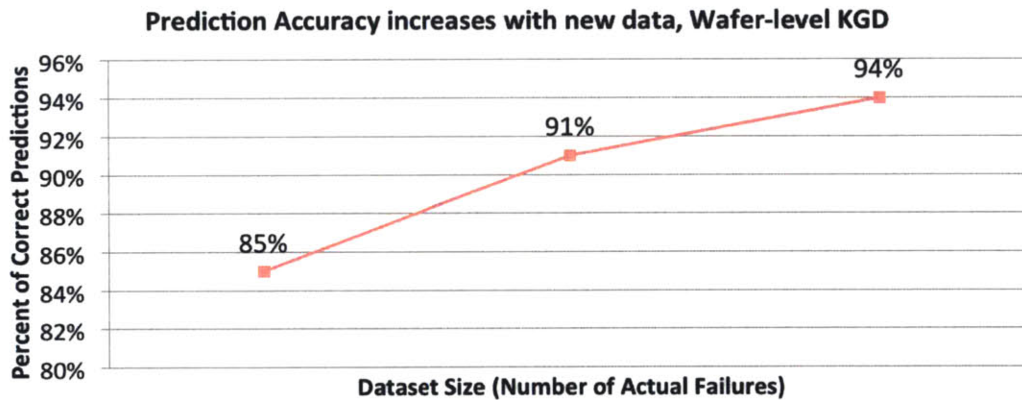


Figure 5-8: Effect of increased number of failure samples in training set

from a subsequent time period. ROC3 is ROC2 plus additional data, similarly sampled from a later time period. The additional data in ROC2 and ROC3 contain both pass and failure samples. The test set is completely independent and remains the same across the three trials. The test set is sampled from the most recent time range after the training set data. The ROC curves in Figure 5-7 show that the predictions improve with each trial.

The wafer-level model results indicate that in future model iterations, a larger data set with an ample number of failure samples is required in order to represent the complexity of the real production data set (Figure 5-8).

The wafer-level model surfaces an additional consideration for prediction modeling. Ag-

gregation to the wafer-level requires that another cutoff threshold needs to be set for wafer predictions. This threshold is the aggregate percentage of die on the wafer that exhibit the same defect. For example, if 20 percent of the die present on a single wafer are predicted to have a specific defect, then the entire wafer should be flagged. This wafer aggregation threshold would need to be derived through data-driven methods over time, keeping cost trade-offs in mind as described in the evaluation metrics section above.

5.3 Die-level KGD prediction model

A die-level KGD prediction model is proposed here that follows the methodology from the wafer-level model, and comparisons are made using different machine learning algorithms. Based on a completely different data set and data extraction process, a new data cleansing procedure is performed manually. After data cleansing, the data set for analysis includes 65 wafer lots, 854 wafers, and 484,739 die. The input parameters are only from SME1 and reflect one day of production data. After filtering on two specific KGD test programs, 170,812 die remain in the data set. A soft bin pareto shows that the top four defect categories capture 85 percent of failures. These top categories are the focus of the analysis, especially the top category which represents almost half the defects.

For the die-level KGD model, several machine learning algorithms are applied to test different methods. Open-source R packages are applied with default arguments in order to compare out of the box algorithm performance without further tuning. This methodology aims to optimize future development efforts by identifying the top machine learning algorithm without fine tuning that would require significant user expertise. It is acknowledged that algorithm performance would vary with further tuning, ensemble methods, or enhanced data refining.

Tree-based algorithms perform the best. The best performing algorithms are shown in Figure 5-9. These are generated using the R packages of *randomforest*, *rpart* for classification and regression trees (CART), *ipred* for bagged classification and regression trees, *gbm* for gradient boosting trees, *MARS* for multivariate adaptive regression splines, and *cubist*. These results are based on the same training and test data sets. The response variable is a

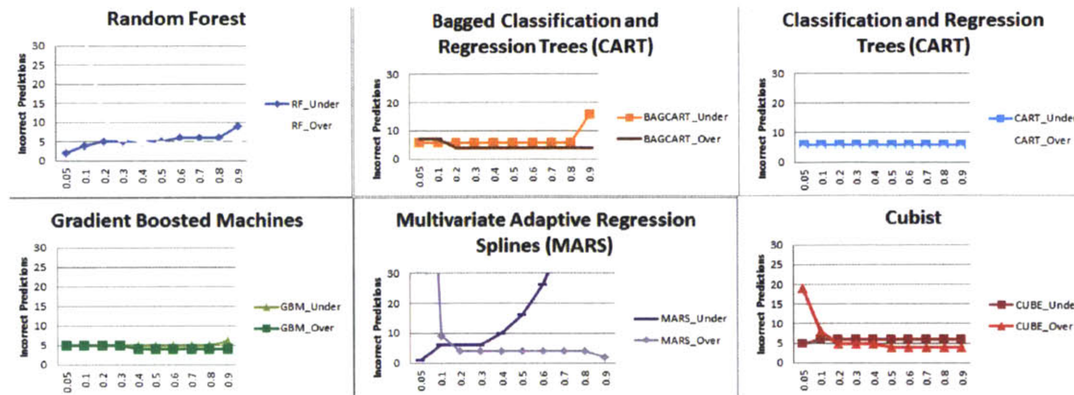


Figure 5-9: Comparison of the best algorithms by threshold vs incorrect predictions

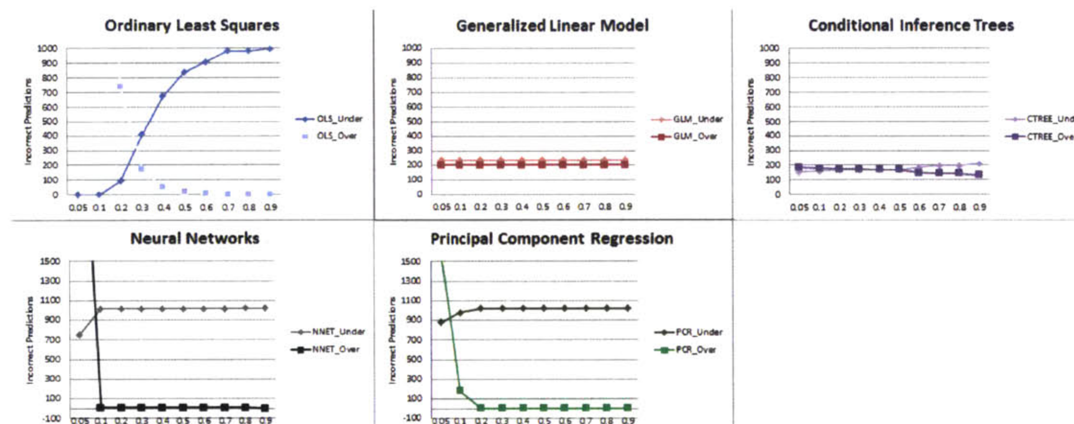


Figure 5-10: Comparison of the worst algorithms by threshold vs incorrect predictions

binary indicator if the particular die belongs in the top KGD defect category.

In Figure 5-9 and 5-10, the x-axis is the threshold and y-axis is the number of incorrect predictions. In random forest, multivariate adaptive regression splines, and cubist, the general trend is a steep decline in overkill (false positives) and a gradual increase in underkill (false negatives). Bagged CART, CART, and gradient boosted machines display a more consistent quantity of prediction errors across thresholds. These results raise the question of a flat prediction error rate across thresholds being preferable since the prediction accuracy would not be as sensitive to changes in the input threshold. For example, if the threshold for the gradient boosted machines model is set to .2 or .8, the prediction performance remains very similar. Whereas with random forest, the number of overkill is high at low thresholds and drops drastically at high thresholds. Yet within a narrow threshold range, random forest performs the best of all the algorithms. Thus, there may exist a trade-off between algorithms

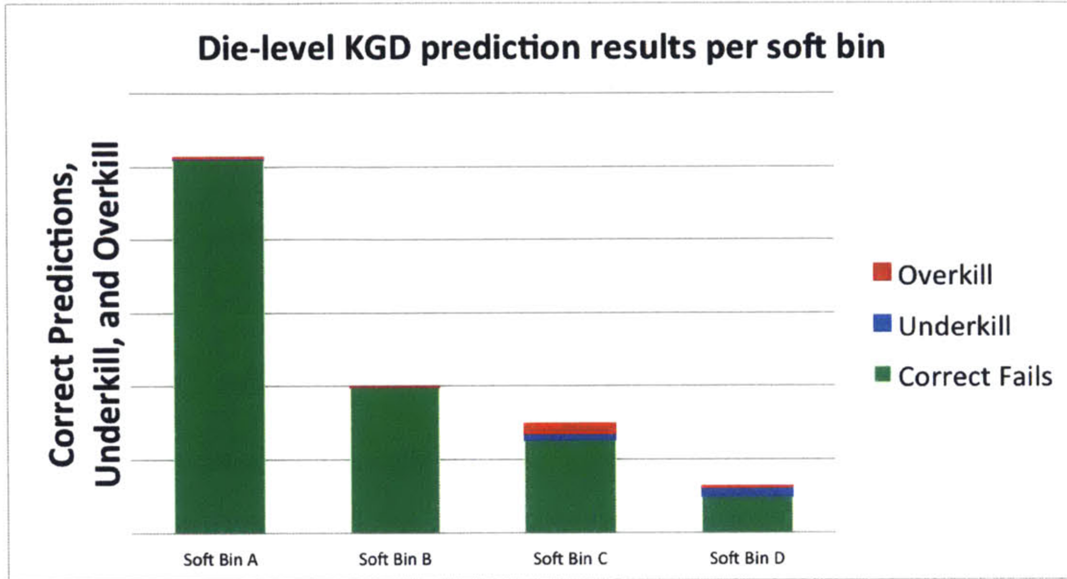


Figure 5-11: Top four KGD soft bin prediction accuracy

that provide consistent prediction results across thresholds versus providing the best solution but at a narrow threshold range. To decrease model implementation complexity, flatter (less volatile) prediction results would be preferable. More exploration into algorithms that can provide both consistent and optimal predictions across thresholds would be the best of both worlds.

Results that represent the worst performing algorithms are shown in Figure 5-10. These R packages include *ols* for ordinary least squares, *glm* for generalized linear regression, *nnet* for neural networks, *pls* for principal component regression, and *party* for conditional decision trees. The number of prediction errors on the y-axis for the top three results in Figure 5-10 reaches 100 and the bottom two results exceed 1000. The scale of incorrect predictions increases by an order of magnitude on average between the best and worst algorithms. More exploration of tuning these algorithms could enhance this model's results, discussed in Section 8.2.1.

The random forest model is the best performer and results in few over and under kill predictions at thresholds between 0.5 and 0.8. Figure 5-11 displays the results of the random forest algorithm on the top four KGD soft bins. Each column represents a different soft bin (defect type) targeted as the response variable. The size of the column represents actual numbers of correct fail predictions (green), underkill (blue), and overkill (red). The values

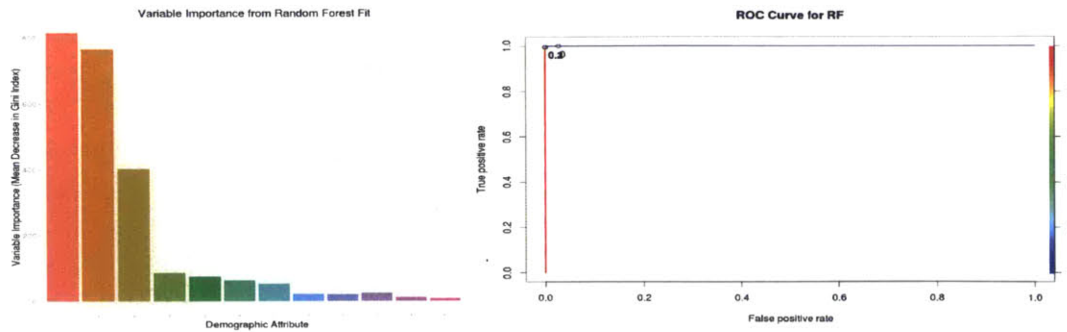


Figure 5-12: Die-level random forest KGD prediction results

are removed from the y-axis and the correct pass prediction volume is not displayed since it would dwarf the other categories. When predicting the largest soft bin "A," this model results in less than one percent of incorrect predictions. "A" describes a specific KGD soft bin that accounts for the largest proportion of defects. "A", "B", "C", "D" are not in order of test program items or in numerical order. They are in order of popularity of defects present in the data set. As the number of fail samples decrease in the training set, the proportion of over and under kill predictions increase. This highlights the need for "big data" without sampling for model training, since the low defect rate limits the number of fail samples available.

In Figure 5-12, the best results are shown from soft bin "A" prediction using random forest in the KGD die-level model. The variable importance plot displays that a handful of input parameters are the most important, and the ROC curve visualizes the high prediction accuracy. Since this die-level KGD data set represents only one day of production, the next model considered seeks to draw data across a broad period of time to test the hypothesis that as time moves forward, predictions will decrease in accuracy.

5.4 Die-level memory test prediction model

The input data set for the die-level memory test model includes SME1, SME2, and KGD parameters along with memory test results merged together by die. The date range is a memory test date between August 30 and October 19, selected in order to gather samples across one and a half months of production. This is a long enough time period such that

a cycle of fab changes is likely to have taken place. After data cleansing, there remain 119 memory test lots, 393 wafer lots, and 924,327 die samples.

The step of response variable selection requires more manipulation than the previous two models, and is described here to inform future research on potential data challenges. The details of these data issues are unique to this data set but represent some of the potential pitfalls associated with data of this nature. The top six FH soft bins represent 80 percent of defects as seen in the other models. Yet these and other memory test FH soft bins require additional expert input to eliminate several data inconsistencies and select a suitable response variable. For example, FH soft bin "X" is the largest defect count observed, but turns out to be assembly-related and had been subsequently moved to another test procedure (not memory test) sometime after October 19. So for the sake of this analysis, all of the "X" defect samples are deleted since they are a data anomaly and would not represent useful predictions. Another top defect "Y" had subsequently been incorporated in upstream checks at the wafer level, but at the time of the data collection was not screened out yet. These represented a package-level defect. Based on expert input, these samples are changed to pass records since it is assumed that these die would have passed the memory test items since test item "Y" occurs at the end of the memory test flow. An additional data cleansing step is employed as well to remove input parameters with identifying information, such as wafer number or die x coordinate. These are assumed to not add value to the model prediction to drive root cause fixes.

Two FH soft bins are selected in this analysis that are "sister errors" in that they represent read after programming for even and odd blocks. Combining the two soft bins is verified in a sub-analysis test described below. As summarized in Figure 5-13, several sub-analyses are used to verify assumptions and provide insights into the complex data set. These are discussed in more detail below.

The first check is to verify if the MT test dates are distributed in a way that represents normal production. Experts confirmed that the test dates are sporadic but a fair representation and that there were no changes in the test program configurations during that time period. Further trials show that limiting the training data to a narrow MT test date does not improve prediction accuracy. It is assumed that the MT test procedure is stable and so no

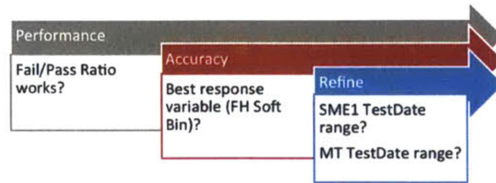


Figure 5-13: MT model development assumptions and data insights approach

matter when the packages were tested they were tested the same way. Yet as for the SME1 test date, it is found that there is a slight improvement in prediction accuracy by limiting the training data to a narrow SME1 test date range. Future analysis could also include a filter on fab version to improve results; further discussion of this appears in Section 8.2.1. Another analysis is to determine which input data set (SME1, SME2, or KGD) has the most important input parameters. In several trials with the CART algorithm and the two selected FH soft bins, SME1 is shown to provide the most significant parameters. As seen in Figure 5-14, KGD has no predictors. SME2 has a minimal number, and SME1 and SME2 have the same predictors as SME1 alone. This analysis narrows down the input data set to only focus on SME1. It is hypothesized that the KGD parameters may need to be refined. Alternatively, this trial could be explained by the fact that current quality processes are actively correcting new defects that correlate between KGD and MT upstream (by updating KGD test programs).

As indicated above, the main algorithms utilized are CART and random forest. Random forest accuracy is much higher but requires more processing power. Computational performance becomes a limiting factor so additional strategies are developed and used here to decrease the data set size. Below describes a sampling strategy to decrease the training set size and still maintain variation across a large time period. The defect rate in the data set is very small, so there exists an imbalanced proportion of pass to fail samples. To keep the valuable fail samples in the training set, only the pass records are randomly sampled. Different ratios of fail to pass records are experimented with. Smaller ratios ensure fast performance but result in extreme fluctuations of overkill at low probabilities (Figure 5-15).

Larger ratios are more accurate but have slow algorithm performance with random forest. For development purposes with manual analysis, sampling the training set pass records shows

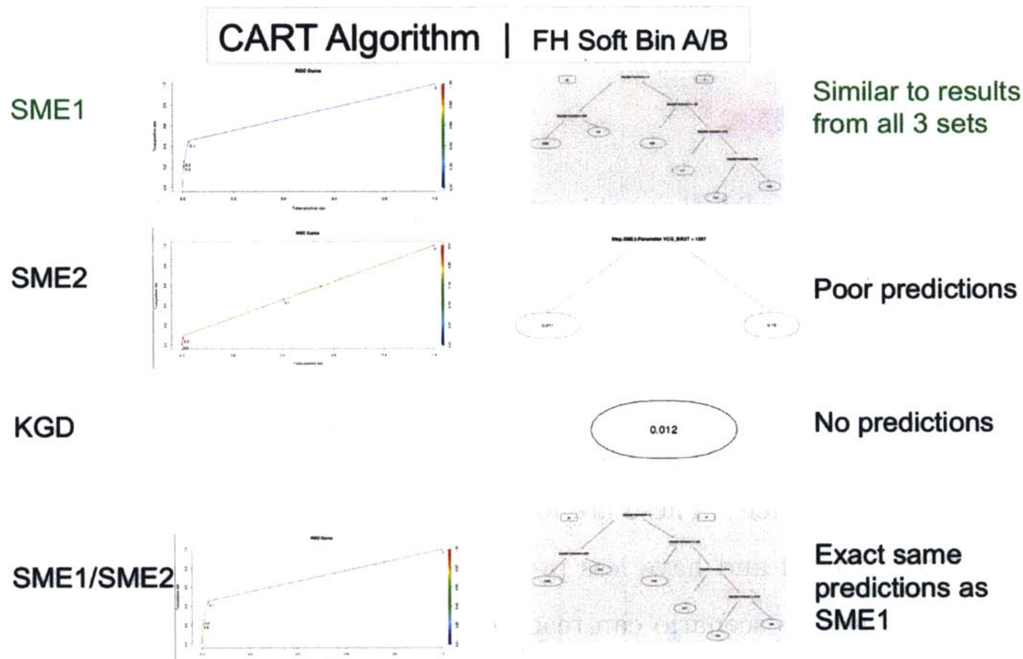


Figure 5-14: Comparison of data sets and important parameters using a CART algorithm

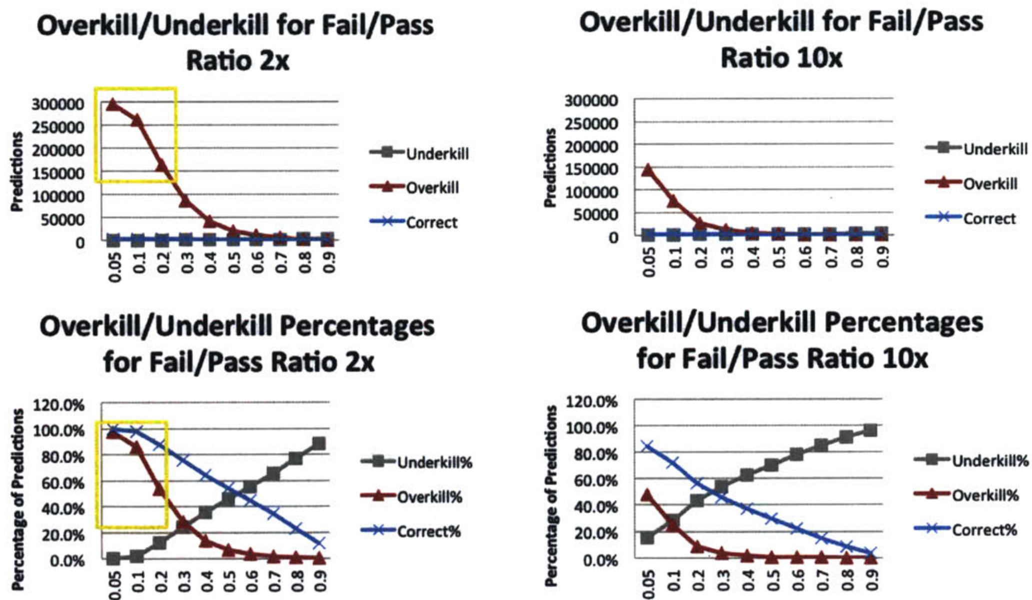


Figure 5-15: Impact of sampling training set pass records

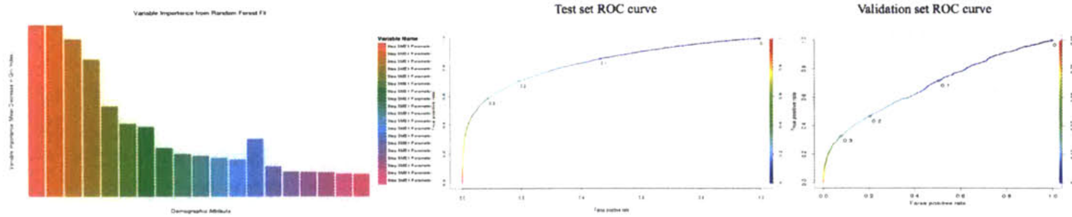


Figure 5-16: Best test and validation set results from MT random forest model, variable importance plot and ROC curves

that the model can still provide meaningful predictions. Sampling pass records can still result in high prediction accuracy, but the threshold range will be more narrow than the non-sampled results. As shown in Figure 5-15, the threshold values from 0.0 to 0.2 are highlighted in yellow on the left. These are for the low fail/pass ratio of 2x. On the right, a higher ratio of 10x is used and have less pronounced fluctuations at the same threshold range. The 2x fail/pass ratio scenario can reach the same prediction accuracy levels as the 10x scenario, but would not be reliable at threshold levels below 0.2. Thus, this sampling strategy is useful for development but if implemented in production, it is recommended the actual production defect rates be represented in the training set data for the highest accuracy possible across all threshold values.

After following the above methodology for data refining, the best random forest results are still inaccurate. In fact, the accuracy for the two FH soft bins is only slightly better than predicting hard bin outcomes (all FH soft bins combined). As seen in Figure 5-16, the ROC curve of the test set (middle) and validation set (right) have low predictive power. The variable importance plot also displays lower importance values on the y-axis (maximum of 200) as compared to Figure 5-12 from the KGD prediction (maximum of 800), further indicating the low predictive power of the input parameters.

Figure 5-17 displays the results from the MT prediction for the test set (top) and validation set (bottom) at the best possible threshold levels (0.9 and 0.8). The results represent the two FH soft bins, only SME1 input parameters, and 5x pass/fail sampling ratio in a training set from 1.5 months of production data. The test set spans the same MT date range (8/30-10/14) as the training data, whereas the validation set spans a subsequent MT date range (10/15-10/19). Some values have been removed from Figure 5-17 but percentages

Test set	
real fails	X
real pass	254934

MT test set - best prediction results						
obs						0.9 threshold
pred	0	1	X1	92.9%	underkill	
0	X0	X1	X2	0.08%	overkill	
1	X2	X3	X3	7.1%	correct defects	
				0.72	Overkill/Correct Defect Ratio	

Validation set	
real fails	Y
real pass	141333

MT validation set - best prediction results						
obs						0.8 threshold
pred	0	1	Y1	98.2%	underkill	
0	Y0	Y1	Y2	0.03%	overkill	
1	Y2	Y3	Y3	1.8%	correct defects	
				2.53	Overkill/Correct Defect Ratio	

Figure 5-17: Best test and validation set results from MT random forest model, confusion matrices

show the prediction accuracy level to be much lower than the prior KGD prediction model. At a 0.9 threshold level, the MT prediction finds only seven percent of defects in the test set. In the validation set at a 0.8 threshold level, the MT prediction finds less than 2 percent of defects. For both test and validation sets, as the threshold decreases under 0.8, the number of false positives exceeds the number of correct defect predictions. Also shown in Figure 5-16, the validation set versus the test set results highlight a drastic decrease in prediction accuracy. The impact of just one week of data highlights the importance of refreshing the model over time, further explored in Section 8.2.2. In summary, the current MT data set and methodology described above do not constitute a successful proof of concept since the results are not accurate enough to make business sense to implement.

A last sub-analysis is performed to simulate the application of the MT prediction model over time. The three trials in Figure 5-18 are from CART algorithm output with the same training set that contains samples with a MT test date between 10/8-10/14. The testing set is the same time frame (10/8-10/14), the validation set is from the subsequent week (10/15-10/19), and the second validation set is from a prior time period (8/30-10/7). Using more extensive historical data for testing is a methodology adopted from Weiss, Dhurandhar, et al. [30]. Comparing the three trials, the test set performs the best and both validation sets decrease in prediction accuracy. At thresholds under 0.7, the number of correct predictions switches from zero to more false positives than correct predictions in the test set in Figure

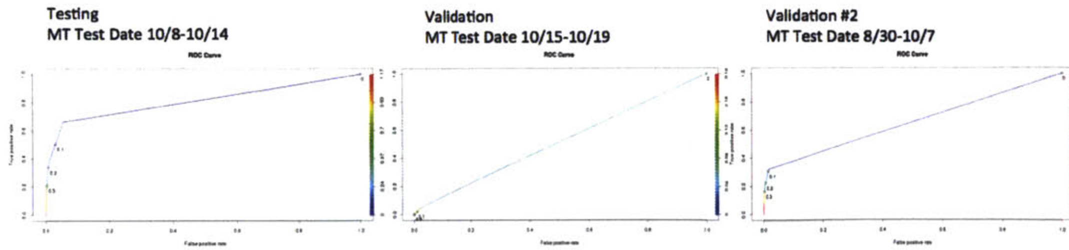


Figure 5-18: MT predictions validation results using CART algorithm

5-18. The validation sets show a more extreme switch than the testing set. This sub-analysis supports the findings shown in Figure 5-16, further indicating that more research needs to be done into keeping a prediction model accurate over time.

5.5 Results Summary

The wafer-level KGD python model (Section 5.2) and the die-level KGD R model (Section 5.3) demonstrate successful proof of concepts in predicting wafer defects. These two KGD models make progress in the development of evaluation methodologies, yet the models themselves are based on limited data sets. The data set and methodology described in Section 5.4 in the die-level memory test R model does not constitute a successful proof of concept. The existing memory test prediction results are not accurate enough to make business sense to implement. Chapter 6 continues on the premise that prediction results can be incorporated into existing screening processes, and provides recommendations how one would implement a prediction model. Chapter 7 uses this prediction model as a case study for a high level strategy how to apply big data across manufacturing processes. Areas for model improvement to increase the accuracy of the die-level memory test predictions are described in Section 8.2.

Chapter 6

Recommendations and Implementation Plan

This section describes several key findings and recommendations based on the successes and challenges from the model and its results. The three top findings discussed in Section 6.1 pertain to performance versus accuracy of high dimensional modeling, the importance of simulating a living model for operations integration, and recommended success metrics measured in overkill/correct defect predictions cost ratio. Implementation recommendations include program changes to the wafer cherry pick process, presented in Section 6.2. More coordination with the wafer fabrication facility would be beneficial to drive root cause fixes, as discussed in Section 6.3.

6.1 Modeling Recommendations

In the experiments and analyses presented here, performance is a key limiting factor to modeling. Several tactics, described in Section 5.4, are utilized to speed up model computation time. First, a faster algorithm is applied for running experimental trials with comparative results. In particular, CART is used in place of random forest and aids in fast iterations of testing hypotheses for data refinement since results for this purpose do not have to be accurate, but rather only need to be evaluated side by side. Secondly, random sampling is used but only for the pass records in the training data set. The training set relies on a

limited number of rare defect samples and so all of these samples remain in the training set. Results show that lower ratios of fail to pass records in the training set could result in optimal results at certain thresholds. Yet at extreme threshold values, incorrect predictions increase at a much more rapid pace (especially overkill). This intuitively makes sense since a training set with many defect samples is over-sampled and biased towards identifying failures in the testing set.

Preliminary results in Section 5.4 show that prediction model accuracy degrades when applied to future time periods. These initial experiments are only with the die-level memory test data. The approach includes maintaining the same training set and running different testing sets. To cut down on variation, all data sets come from the same fab processing week (the same SME1 week date range) and go through the same data refinement process described earlier in Section 5.4. The recommendation is to keep separate teams (fab process engineers, test engineers, and data owners) across locations in close coordination since fab processing and the cherry pick wafer sort processes are both constantly changing over time. Secondly, if one is evaluating new modeling techniques or technologies provided by vendors, one should request that the modeling be simulated over time. This can be performed in the same manner as described in Section 5.4 by using historical data and only the knowledge available at the time to re-create a full operations implementation. As described in Chapter 8, further real-life simulations can be expanded by removing failure records that a prediction model would have captured to see the model's long term impact on operations and anticipate potential challenges.

A proposal is made here as to how to calculate evaluation metrics that fit business needs in a straightforward manner. Since overkill percentage is very low given the small defect rate, it is important to evaluate the model based on the numbers of defects (predicted and actual) and to generally disregard percentages of prediction errors. In order for a prediction model to be implemented, an experienced analyst will need to provide the business with prediction thresholds. A simple implementation would keep the initial threshold the same over time. A more advanced implementation would update the threshold regularly to re-balance false positives to true positives according to the cost ratio proposed in Section 5.1.2. Also explored in Section 5.3, it is recommended to select an algorithm that exhibits a flat prediction error

rate across thresholds to simplify model implementation and maintenance over time.

In order to set the thresholds and fully evaluate the business impact of a prediction model, financial inputs and thresholds will need to be set. These financial averages vary by product, process, and time. The methodology introduced in Section 5.1.2 of a "overkill/correct defect ratio" requires the estimation of the revenue lost per false positive (overkill) and the cost savings of predicting a defect (correct defect prediction). Even if these exact values are not known, one can estimate a comparative ratio. Then, when evaluating a potential prediction model or algorithm, the input threshold can be narrowed down to a range where this cost ratio is met. If the cost ratio cannot be satisfied at any threshold, then the model should not be implemented.

Consider the following example, assume there is a one dollar cost savings in assembly and testing costs for predicting a defective die, and there is a ten dollar overkill cost to degrading a valuable die and losing potential revenue. Then one would prefer to only implement a model at a threshold where the "overkill/correct defect" ratio is less than 0.1. Referring back to the confusion matrices in Chapter 5, the ratio of 0.048 in Figure 5-2 would meet this criteria. It follows that a threshold level of 0.8 would be an acceptable input for the prediction model implementation. In contrast, the confusion matrices in Figure 5-17 result in ratios of 0.72 and 2.53. These do not meet the 0.1 cost ratio requirement so this prediction model would not be acceptable. In this example, the threshold level can be set to less than 0.1 in order to further minimize false positives. If the business is risk-averse about lost revenue, one would select a lower threshold that would decrease false positives (and increase false negatives).

The proposed methodology to find an acceptable threshold as a model input that makes business sense to implement and limits overkill is as follows. X_{costs} represents the "overkill/correct defect" ratio based on estimated costs. Assume overkill is more costly than underkill and there exists a trade-off relationship between overkill and underkill. The purpose of this methodology is to result in the selection of a threshold that maximizes correct defect predictions and minimizes underkill within an acceptable overkill proportion.

1. Calculate the prediction model's ratios across all thresholds (X_{model} = overkill/correct predictions from the test set results).

2. Find thresholds where X_{model} is less than X_{costs}. If no thresholds exist where this is satisfied, then the prediction model does not make sense to implement as is. If a threshold range does exist,

3. Select the lowest threshold in the range.

In terms of a strategy for incorporating a new big data platform into existing data systems, the recommendation is to leverage the new platform as a development and experimentation tool. The existing data repository can implement proven solutions developed on the new platform. Examples of implementing a proven solution include vendor customizations or automation on the existing data systems. Thus, the existing data system remains the single, long-term system of record and production processes remain undisturbed on the existing system during development projects. Meanwhile, a new big data platform is ideal for experimentation with new algorithms and would provide new tools to solve existing problems. Therefore, the combination of the existing and new big data platforms is complementary.

6.2 Cherry Pick program

Cherry pick program changes would require updates to the existing IT and data flow systems. It is recommended to only incorporate cherry pick changes that have been validated over time with historical data and meet business needs. The evaluation of business impact could be calculated using an avoided scrap unit cost model. A visual of where the prediction model can be incorporated into the existing process is pictured in Figure 6-1 and the current cherry pick process is described in Section 4.1.2.

For implementation, the prediction model results would need to be calculated real-time after the fab test data is generated (SME1 and SME2). The prediction results could be fed into the cherry pick process and make alterations to the wafer map. It is recommended to have manual configuration and oversight by experts in order to take into account external changes in supply and demand in the market. For example, balancing wafer supply across multiple product lines or anticipated changes in customer demand. Cross-functional team coordination is also recommended to collect these diverse sources of information about

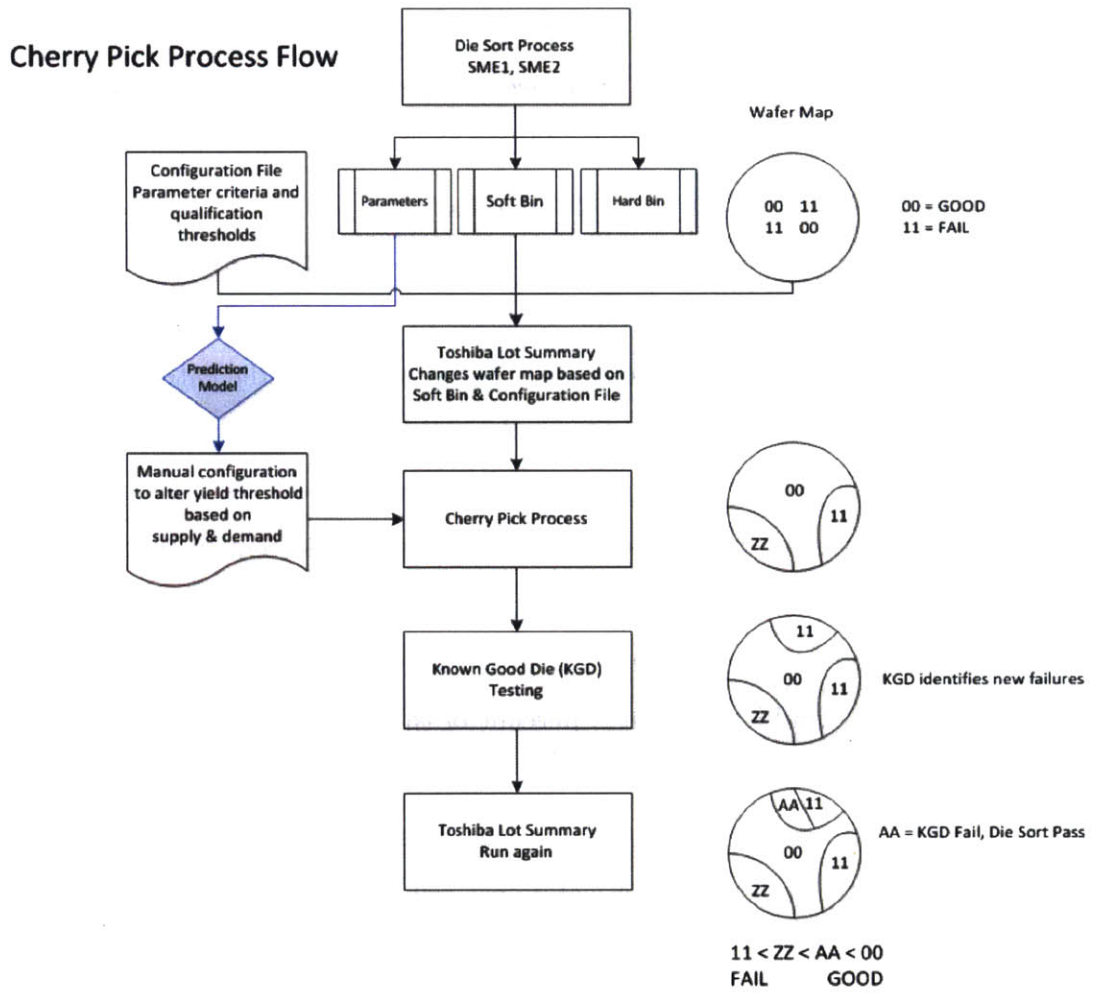


Figure 6-1: Prediction results incorporated into decision-making process

changes in supply and customer demand across the company. In terms of calculation time, the prediction results need to compute and update the wafer map file in less than a day to ensure there is no delay in the cherry pick process.

To implement the KGD prediction model, it is recommended that model results be incorporated during the cherry pick process. Wafers that the prediction model identifies as exceeding the allowable portion of die with defects (yield) would be sorted into a non-prime (lower quality) product process flow. In the non-prime test process flow, wafers undergo a less stringent KGD test program. Thus, yield would increase on the overall prime product KGD test and the company would avoid non-value added testing time on the 24/7 KGD test machines (stringent test programs need longer testing time).

An example to explain Figure 6-1 and the impact of a prediction model implementation is described here. Assume that the allowable die yield percent per wafer is set at 95 percent for prime wafers. If there is a wafer that currently has 96 percent of die categorized "00" (die sort pass), two percent as "ZZ" (non-prime die defined by cherry pick configuration rules) and two percent as "11" (die sort failures identified in the fab test results), then this wafer would continue to KGD testing for prime wafers. Assume two percent of this wafer will fail KGD testing for a specific defect seen before but not accounted for in the cherry pick rules yet. With the current process, the KGD testing would result in 94 percent yield. In Figure 6-1, the portion of "AA" would represent 2 percent of the die that were a die sort pass but a subsequent KGD fail. In retrospect, this wafer should have been downgraded and tested with a non-prime KGD program where it would have passed the less stringent test criteria.

We cannot retrospectively change the wafer sort but we can use a prediction model to effectively add foresight of the KGD results to update the cherry pick process. If a real-time prediction model is trained for this specific defect, then the cherry pick process could be updated before the KGD test program begins. In this example, the prediction model would predict the two percent of anticipated KGD failures. The portion of "ZZ" would increase by two percent and the total wafer yield would be 94 percent. The wafer would not pass the allowable yield of 95 percent and automatically be downgraded to a non-prime product and undergo the less stringent KGD test program.

If the prediction yield percent does not decrease below the allowable yield there are still

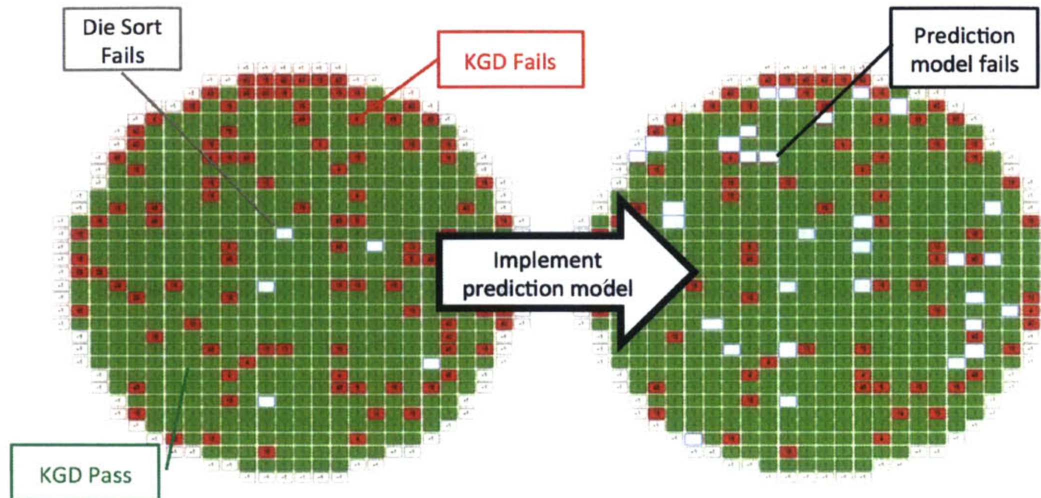


Figure 6-2: Mockup of updated wafer map based on prediction model results

benefits to the prediction model. The portion of "ZZ" would increase and thus reduce the portion of "AA" identified by KGD. This would decrease test time and provide a more accurate overall yield of the die sort process that can aid other decision-making processes.

Another way to visualize the prediction model implementation changes is shown in Figure 6-2 as a mock up of updates to the wafer mapping file. The wafer mapping file is a structured data file but can be visualized as a map, shown in Figure 6-2 and also displayed earlier in Figure 2-7. Structured output from a prediction model could update the rules in the cherry pick program. Following the example from Figure 6-1, the cherry pick program updates would incorporate prediction model results and update values of "00" to "ZZ" for specific die identified by the prediction model. As seen in Figure 6-2, the red die represent "AA" (die sort pass but subsequent KGD fails). The white die represent die sort fails, and green represent both die sort and KGD passes. The wafer on the left is how the wafer map would appear with the current process after KGD testing finishes. The wafer on the right represents how the wafer map would appear if the prediction model were implemented. A portion of the red KGD fails would effectively become white prediction model fails. Manipulating the input wafer map file would enable the current KGD program process to function as normal. The KGD test process is a critical production step so keeping this process as is would allow for a smoother implementation versus an implementation solution that requires a complete overhaul of the current process.

The next implementation step would be to apply the memory test prediction results to update the wafer mapping file. More research can determine if the KGD test parameter inputs add predictive power to the memory test prediction. If so, then changes to the wafer mapping file would occur after KGD testing is complete and initiates the computation of the memory test prediction model. Changes to the manufacturing process would physically occur during the die attach step of assembly. In this step, the system refers to the wafer mapping file to determine die placement on each product and die stack. If the die is predicted to fail memory test, then the die can be placed on a non-prime product or stacked in such a way that the other die in the stack can compensate so that the overall product performance still meets customer needs. If more research reveals that KGD test parameter inputs do not add predictive power to the memory test prediction, then the memory test results can be incorporated the same way the KGD prediction results are in Figure 6-1. It is recommended to keep the prediction models on a die-level granularity in order to simplify implementation for a combined KGD and memory test wafer mapping file update scenario.

6.3 Wafer fab

The output from a prediction model would ideally convey an automatic report of which die sort parameters are most important in defect prediction back to the wafer fab. Combined with the wafer fab engineering knowledge, this die sort parameter information can drive root cause investigations into specific machines, materials, and processes. One concern is that tweaking one process step within the wafer fab could impact other processes unexpectedly. Another concern is the cost of these changes. A recommendation for engaging the wafer fab engineers is to create a summary report that highlights the change in yields between the regular die sort and prediction model die sort results.

Another recommendation is to organize the prediction model results by failure mode (program, erase, etc.) to align more with the wafer fab root cause investigation process. Additionally, aggregating the prediction results at the wafer or lot level (versus die-level) would provide more meaningful feedback since currently fab analysis exists at the wafer or lot level. Wafer fab engineers usually investigate wafers or lots by median or average in order

to determine process engineering changes. Lastly, the prediction model input data should be separated by fab. Separating input data by fab is a modeling next step described in Section 8.2.1. Calculation of the savings from the wafer fabrication root cause improvements would require a custom financial model. Implementing fab improvements is challenging but also anticipated to have greater financial benefits than the avoided cost of assembly defects in KGD and memory test.

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Chapter 7

Strategy for Applying Big Data and Advanced Analytics to Semiconductor Manufacturing

As introduced in Chapter 3, some data-related challenges are longstanding in semiconductor manufacturing while other opportunities arise due to the advent of big data and new advanced analytic capabilities. One main roadblock is transitioning from traditional, separated data stores into a consolidated data warehouse. Semiconductor manufacturing equipment and information technology systems were not historically designed to facilitate large quantities of real time data that can easily trace and integrate all aspects of and inputs to the process (data traceability). Thus, root cause analysis is reactive and predictive modeling has infrastructure upgrade challenges in order to obtain and refine data. With the arrival of big data methods, domain knowledge that traditionally lies within specialized engineering teams requires partnership with data scientists. With heavy reliance on data in models such as predictive modeling, a major challenge is data quality of the expansive data sets that may have missing values and/or require cleansing or logic to be useful. Lastly, the larger problem sets face scaling and computing speed issues.

This chapter focuses on a proposed methodology to frame big data challenges as opportunities. The following sections have a generalized approach for evaluating problem sets and potential solutions. Section 7.1 describes a decision framework for identifying, categorizing,

and evaluating major problem sets present in a manufacturing organization. The framework aims to place problem sets within the best solution architecture of an enhanced platform, toolset, or a big data ecosystem based on user needs and data structure. Section 7.2 proposes a high level methodology how to assess the company impact of addressing these major problem sets in terms of financial and customer benefits.

7.1 Framework to Evaluate Applicable Problem Sets

The first step is to map out the existing problem sets across the manufacturing process. Problem sets are defined as areas for process or quality improvements anywhere in the manufacturing process. Problem sets can range from broad to specific challenges, such as wasted time and effort when new products are over-qualified (over-tested) or a specific, known failure mode like bit line short-circuits. Some problem sets may currently be unknown unknowns, so the priority is to first focus on the known issues that cause the largest number of operational issues. Along with the definition of the problem set, a survey of the current and ideal state are necessary to evaluate each item. Categorizing the location, process, and impact category is helpful for defining the solution options. Examples of areas to explore to identify the main problem sets include: initial chip design specifications, fabrication process data, assembly process data, and solid state drive-level process and testing.

The decision framework pictured in Figure 7-1 is proposed to map a problem set to potential big data system recommendations. The problem set is the input at the top and decisions flow to the bottom. Each decision point requires an evaluation by both the domain and data science experts. The first step is data access to make sure the analysis is feasible at all (Do we own the data? Do we need to upgrade machinery or information systems to collect the necessary data?). The second step is data structure. Traditional relational databases cannot handle unstructured data, such as images, text-heavy, or irregularly formatted information. For systematic analysis, these problem sets are best addressed with a big data ecosystem. Semi-structured data may fall into either category depending on expert input. The next step is evaluating the volume of data taking into account file size and retention. The main questions to ask for this step are about the dimensionality of the data (are there complex,

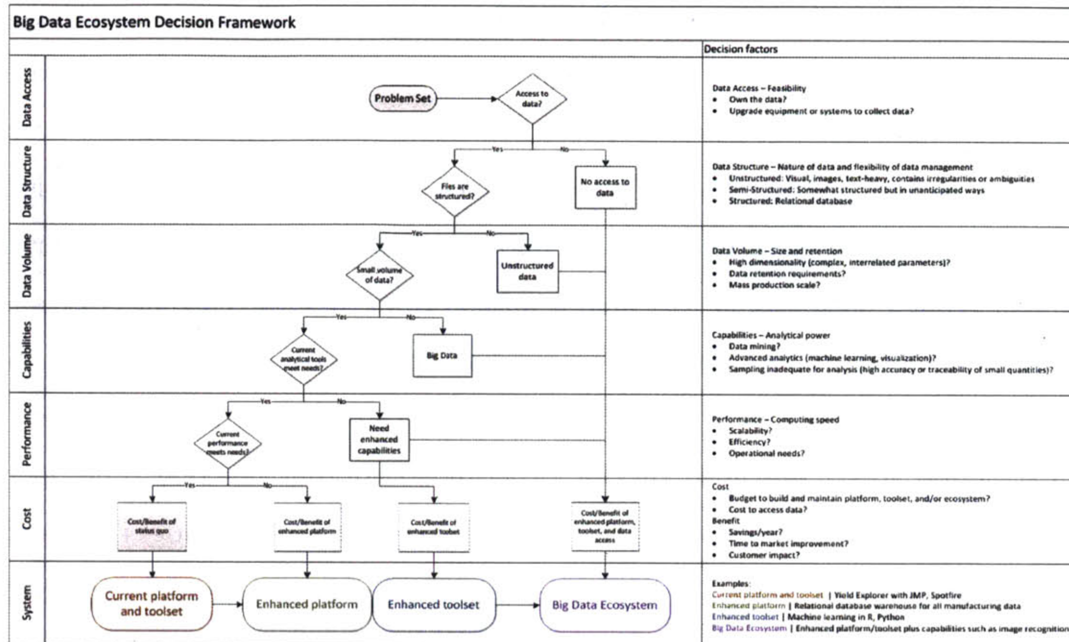


Figure 7-1: Decision framework for evaluating big data ecosystem opportunities

interrelated parameters?), does the analysis require data over long periods of time, and if the analysis is at mass production scale. If so, this is a fit for a big data ecosystem. The next step is to evaluate if analytical capabilities are missing in the current state. Advanced analytics like machine learning, visualization, data mining, or accurate trace-ability of small quantities (when sampled data will be inadequate) are examples of additional capabilities that might be considered. If enhanced capabilities are desired, a big data ecosystem or an enhanced toolset would be options. If capabilities do exist, then the last question is performance or computing speed of the current state. If the current state cannot scale or is not efficient enough to meet operational needs, then an enhanced platform or big data ecosystem would be options.

After the categorization is complete, the final choice will depend on cost/benefit analysis. For example, if the only issue is performance then the status quo can be compared to the enhanced platform or big data ecosystem to determine what options to select. Similarly, a lack of capabilities could be addressed with a wider toolset or big data ecosystem. An example of an enhanced platform is a relational database that consolidates all manufacturing data instead of storing in separate data shares. An example of an enhanced toolset is machine learning algorithms from R or python. A big data ecosystem might include the above and

contribute additional capabilities such as image recognition and efficient data storage.

Using the prediction model developed in this thesis as a case study, the following describes how the problem set maps to system improvement recommendations using the framework proposed above and pictured in Figure 7-1. In this case, access to the data did exist. Files were structured in nature. The model was a pilot proof of concept and only experimented with sampled, refined data sets of less than a million rows. Yet the current state's analytical tools did not meet the user needs. The existing data repository did not have built-in capabilities to enable machine learning or customized data refinement. As for performance, the data extraction speed was adequate for experimentation but not real time operations due to delayed data transfer. Following the decision framework, this model requires an enhanced toolset/platform or a big data ecosystem to enable use on a production scale. The investment decision depends on the cost benefit analysis of each solution described below.

7.2 Impact Assessment Methodology

After surveying the problem sets and evaluating their current state using the decision framework and expert/user input, the next part of the assessment methodology is to determine cost and benefit figures. Evaluating the cost of the upgraded or new system could entail working with a wide variety of groups and exploring many technological solutions. For example, vendor or internal solutions or enhancements might be considered and compared. The cost components to consider include but are not limited to: build, test, integrate, maintenance over time, and data access infrastructure. To measure the benefit of the solution options would require collaboration with internal teams to estimate the financial payback and other important company benefits. Some benefits include cost savings per year, time to market advantages (converted to financial savings if possible), and customer impact (qualitative and case studies of what-if scenarios if previous customer issues had been averted).

An example visualizing a map of problem sets to opportunities is shown in Figure 7-2. The potential solution options (platform, toolset) is along the x axis in order of increasing complexity that matches the decision framework. The y axis features a qualitative estimation of overall company impact (net benefits), increasing upwards. The bubbles represent each

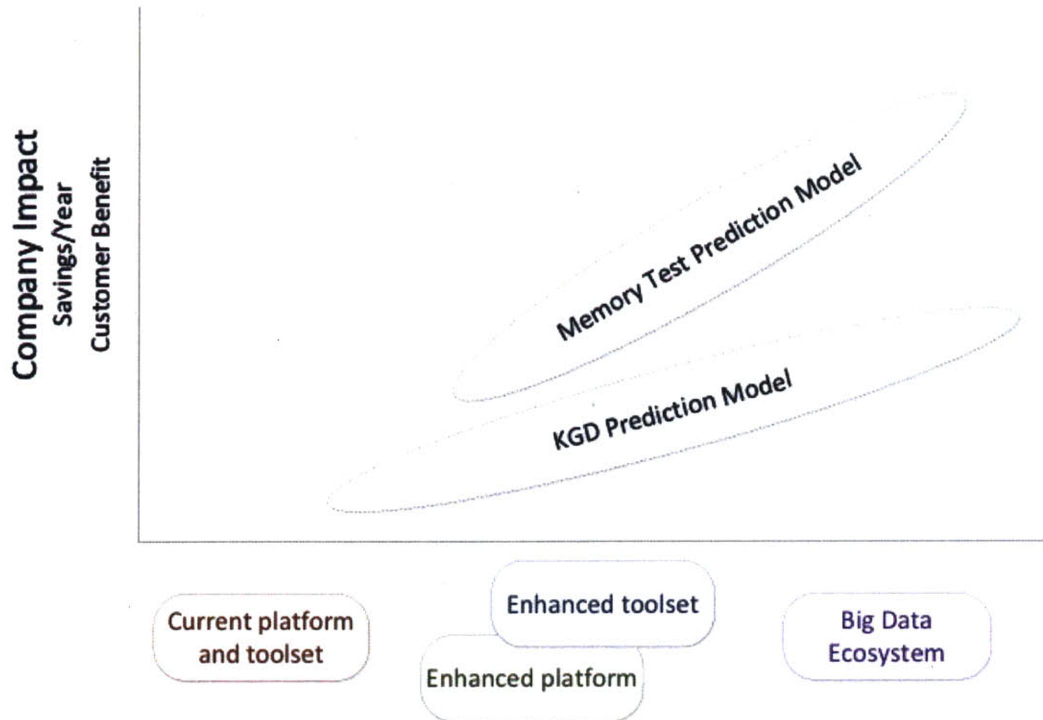


Figure 7-2: Framework to map problem sets to opportunities

problem set. They span multiple solution options but are anticipated to have a higher customer impact with more robust systems that scale and unlock additional capabilities over time. In this case study, the memory test prediction model has higher impact to the company so it is above the KGD prediction model. The memory test prediction model also slopes upward more steeply since the complexity and data challenges inherent in this problem set require and would be more effectively addressed with a big data ecosystem rather than enhancements to the current systems.

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Chapter 8

Conclusion

The research and prediction modeling described in this thesis represent a promising new direction for SanDisk and other semiconductor companies to apply big data and advanced analytics to improve their manufacturing operations. As described in Chapter 2, the flash memory industry is a highly competitive space with technology that improves faster than Moore's law. Big data applications to semiconductor manufacturing is an open area of industry and academic research currently and recent publications are described in Chapter 3. The existing high-tech, complex semiconductor manufacturing process, covered in Chapter 4, generates detailed quality data which can be leveraged in a big data environment. Results from three prediction models in Chapter 5 demonstrate the potential for machine learning methodologies to unlock new ways to approach and solve traditional semiconductor manufacturing challenges. Recommendations are proposed in Chapters 6 how to implement prediction models to improve the current manufacturing and data system processes. These modeling results serve as a case study in Chapter 7 that is proposed as part of a high-level, expansive strategy to apply big data and advanced analytics capabilities to drive cost savings across the manufacturing organization.

Chapter 8 summarizes modeling results, limitations, and next steps to enhance the prediction models described in this thesis. As described in Section 8.1, this pilot analysis demonstrates that there exists a potential opportunity to improve the KGD wafer sort process. Yet the memory test prediction model requires more development in order to achieve high enough predictive accuracy to merit implementation. In terms of key findings, limita-

tions have been identified pertaining to the computational performance of high dimensional modeling and are discussed in Section 8.1. Accuracy of model predictions and speed of computation are trade-offs, especially with big data sets. Section 8.2 describes several areas to improve the model, including enhancing the data set, validating the model over time, and refining processes that tie into the prediction model.

8.1 Results and Limitations of Model

The wafer-level KGD prediction python model with random forest classifier is found to be a successful proof of concept, demonstrating that machine learning can successfully predict a wafer defect. The development and analysis methodology established has been expanded and attempts made to apply to die-level memory test modeling. The model is found to be limited by the provided data set, which is a small, manually selected sample that does not reflect true production level volumes.

The die-level KGD prediction R model has demonstrated a successful proof of concept and advances the methodologies established in the successful wafer-level model. Processes for data extraction, refining, defect selection, and evaluation metrics have been created. Yet the model is found to be limited by the provided data set of one day of production data. Initial findings show that tree-based algorithms are the most accurate, though there exist opportunities to further tune other algorithms to improve performance since only default settings were utilized.

A die-level memory test prediction R model has been developed following the same methodologies as the previous models, but have not resulted in a successful proof of concept. The memory test data spans a longer time period (1.5 months) and includes more parameter data inputs, and thus required comprehensive refining and analysis. Several sub-analyses have been performed to provide insights into this complicated data set; new findings and methodologies to be applied to future models have been proposed. The primary algorithms utilized here are CART and random forest; neither are accurate enough to predict FH soft bin failures to satisfy business needs. The model is believed to be limited by data access issues which are explored further in Section 8.2.

8.2 Recommendations for Next Steps for Model

Many ideas have been generated to enhance and expand the wafer and die-level prediction models. Section 8.2.1 describes ideas about how to enhance the data set for the memory test prediction model to improve accuracy. Section 8.2.2 provides thoughts on how to simulate the prediction model over time, which is an area of primary concern for production implementation. In Section 8.2.3, related processes are listed that could be improvement projects on their own or lend additional insight to the prediction models.

8.2.1 Enhance Model Data Sources and Logic

The first improvement to the memory test data set would be to enhance the granularity of the response variable. Currently the response variable is binary for the FH soft bin. A continuous response variable would be expected to greatly improve the accuracy of the predictions. At the time of this model, a continuous response variable did not exist in the data set. Yet with further alterations to the data feeds, a better response variable may be obtained. One idea is to make the response variable the number of bad blocks and addresses. An interim step would be to simulate continuous data in order to demonstrate the potential accuracy improvement over the current binary response variable. This synthetic analysis could be an immediate next step to make the case for investing in an enhancement to the current data feed.

A second next step would be to enhance the model logic. The current model does not include spatial inputs, such as die location at center or edge of wafer. Secondly, another enhancement would be to aggregate the die by lot and wafer in the input data to show the relationship between these groupings. Plane level was not explored in this model, and is another avenue to obtain further data granularity. Lastly, separate analysis by fabrication facility and KGD test program could be considered. In the existing memory test analysis all test results from multiple fabs were combined together and all KGD test programs were combined and assumed to be equivalent. These assumptions are potential flaws in the model logic, and experimentation could explore if these entities can be combined or should be kept separate.

Another future step would be to tune the algorithms and apply ensemble methods. For example, one could use random forest to narrow down to the top input parameters (perhaps 20 or 30) and then use logistic regression with the vastly decreased data set. The random forest algorithm can be further tuned itself, especially with respect to the number of trees, minimum observations, and depth.

Another important enhancement to the prediction model would be to increase the sources of input parameters. Currently assembly factors were not included. Yet it is possible that assembly processing may play a role in memory performance and defects. Although there exists a separate final test for assembly-caused errors, including assembly data would test this hypothesis. Last but not least, adding fab inline data would be expected to greatly improve prediction accuracy. This data is currently unavailable for analysis.

8.2.2 Simulation of Operational Model over Time

As discussed in Section 6.1, it is important to further validate the process of intelligently keeping the model accurate over time before making changes in production. The model is likely to need to evolve over time since the process is constantly changing as is. Indeed, the model could trigger root cause fixes at the fab based on the model output. Depending on whether or not it is financially and technically feasible to address specific root causes at the fabrication facility, the expected implementation fixes may occur at the fab, or may create new cherry pick wafer sort improvements. Either way, the prediction model will have to chase a moving target and will need a mechanism to automatically re-train itself. One proposal would be to start with a simple moving time period to cut off the training data set as a baseline, and experiment with Bayesian fusion methods that naturally updates and removes stale training data [34].

Another important aspect of the implementation of a continuously re-trained prediction model is that removing predicted failures from the population means that fewer eventual product failures are available to re-train the model with. An expected downside of implementing a prediction model and die-selection decision procedures is that over time the training data would become outdated and could result in unnecessary overkill. Without new training data containing failure samples to re-train the model, there would not exist

a method to verify the prediction accuracy as time passed. One strategy to address this challenge would be to create a mechanism to allow predicted die failures to continue through normal processing. These expected failures can be used to verify that the model remains accurate over time, but can be allowed in sufficiently small volumes to minimize lost revenue. Sensitivity analysis and simulation can be performed to test the impact and cost implications of this testing of the boundary conditions.

Due to the high volume of each product and number of distinct product lines, an implementation of prediction models for all products would require robust automation and an intuitive monitoring interface. All processes involved from data retrieval, re-training of the model, updating thresholds, and feedback reports to the fab and cherry pick process would need to process large amounts of data continuously to keep up with operational needs. It would be expected that monitoring processes and analysts would oversee an interface that alerts on irregularities in the prediction model results, similar to the statistical process control procedures already in place on the manufacturing line.

8.2.3 Optimize Related Sub-processes

In this complex manufacturing process there are several adjunct areas that offer opportunities for further exploration and process improvements. The processes described below are areas for deeper analysis that could provide cost savings to the company on their own or support the results of the prediction model.

The KGD and memory test programs are optimized for speed to decrease expensive fixed capital investments of test equipment. In general, once a die is determined to reach the failure threshold during its test program the testing procedure terminates. One area for further exploration would be to run a full engineering test program that does not terminate when the failure threshold is reached in order to gain insight to correlating multiple failure modes (soft bins). This would be valuable for both KGD and memory test. Additionally, memory test programming does not necessarily align with the die sort or KGD test programs. A full engineering test program analysis could aid future correlation studies of the entire test flow results between die sort, KGD, and memory test.

A related area of cost savings would be to simulate the impact of reducing test steps in

KGD or memory test or sampling. Currently all die and products are tested. There would be time and cost savings if a prediction model identified die that are expected to pass or fail and then the test program itself could reduce the applicable test steps. If the expected pass and fails are separated, the remaining die could be randomly sampled for specific tests for cost savings from fixed asset reduction.

The concept of identifying high performing die has additional financial benefits. This project focused on predicted die that would not perform as expected. On the other hand, identifying die that could perform even better than expected and pass more stringent test programs would positively impact revenue. Currently these die may be hidden among the production data but there exists no signal to identify their existence. Thus, design of experiments or a sampling strategy with a separate test flow would need to be implemented to purposely test these predicted failures on the boundary condition. This is related to the ideas set forth in Section 8.2.2 to maintain a living model over time, but taking a step further to recognize the financial benefits of predicting unexpected high performing die.

The methodologies identified in this project for predicting wafer defects can also be expanded to upstream and downstream manufacturing steps. A similar methodology can be applied to assembly errors, which would incorporate additional sets of data inputs, such as the materials provided by suppliers, machines, environment metrics, human labor, and in line processing data. An example of the prediction model response variable would be die cracks. Another area to expand the prediction model would be to the downstream process of solid state drive (SSD) manufacturing. Aligning all upstream data to predicted SSD quality results would drive cost savings and coordination between facilities.

Bibliography

[1] Channel News Asia. "Singapore competition watchdog clears Western Digital's acquisition of SanDisk", 2016. http://www.channelnewsasia.com/news/business/singapore/singapore-competition/2440402.html, accessed February 2016.

[2] Duane Boning. 2.830J / 6.780J / ESD.63J, Control of Manufacturing Processes (SMA 6303), (Massachusetts Institute of Technology: MIT OpenCourseWare), http://ocw.mit.edu (Accessed June 1, 2015). License: Creative Commons BY-NC-SA. Spring 2008. Accessed June 2015.

[3] Leo Breiman and Adele Cutler. CRAN R-Project package RandomForest, 2014. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf, accessed August 2015.

[4] Chen-Fu Chien and Shih-Chung Chuang. A Framework for Root Cause Detection of Sub-Batch Processing System for Semiconductor Manufacturing Big Data Analytics. *Semiconductor Manufacturing, IEEE Transactions*, 27(4):475–488, 2014.

[5] Raman Chitkara. "Mobile Technologies Index Storage: Quenching the thirst for more", 2012. http://www.pwc.com/gx/en/industries/technology/mobile-innovation/mobile-storage-quenching-the-thirst-for-more.html, accessed February 2016.

[6] Kevin Conley. "Flash: The Great Disruptor, Flash Memory Summit 2015", 2015. http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2015/20150811_Keynote4_Conley.pdf, accessed February 2016.

[7] Michael Hackerott. "Semiconductor Manufacturing and Engineering Data Analysis", 2004. https://sites.google.com/a/informatic-solutions-llc.com/informatic-solutions-llc/home/semiconductor-informatics, accessed June 2015.

[8] R. Hattori. "Big Data and analytics for semiconductor manufacturing", 2013. http://www.semiconjapan.org/en/sites/semiconjapan.org/files/docs/SPR9_35_IBM_Hattori%20Ryuichiro.pdf, accessed November 2015.

[9] Shao-Chung Hsu and Chen-Fu Chien. Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *International Journal of Production Economics*, 107(1):88–103, 2007.

[10] McKinsey Inc. "Big data and the opportunities it creates for semiconductor players", 2012. http://www.mckinsey.com, accessed August 2015.

[11] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R.* Springer Publishing Company, Inc., 2014.

[12] Seokho Kang, Sungzoon Cho, Daewoong An, and Jaeyoung Rim. Using Wafer Map Features to Better Predict Die-level Failures in Final Test. *Semiconductor Manufacturing, IEEE Transactions*, 28(3):431–437, 2015.

[13] Dana Cheree Krueger. *Semiconductor yield modeling using generalized linear models.* PhD thesis, Arizona State University, 2011.

[14] Nathan Kupp and Yiorgos Makris. Integrated optimization of semiconductor manufacturing: A machine learning approach. In *Test Conference (ITC), 2012 IEEE International*, pages 1–10. IEEE, 2012.

[15] Robert C Leachman. Yield Modeling and Analysis, IEOR 130, Methods of Manufacturing Improvement, UC Berkeley. Spring 2014.

[16] Benjamin Lenz and Bernd Barak. Data mining and support vector regression machine learning in semiconductor manufacturing to improve virtual metrology. In *System Sciences (HICSS), 2013 46th Hawaii International Conference*, pages 3447–3456. IEEE, 2013.

[17] Sathyan Munirathinam and B. Ramadoss. Big Data Predictive Analtyics for Proactive Semiconductor Equipment Maintenance: A Review. *Academy of Science and Engineering (ASE), USA, © ASE*, 2014.

[18] Seung Hwan Park, Cheng-Sool Park, Jun Seok Kim, Youngji Yoo, Daewoong An, and Jun-Geol Baek. Pattern Recognition Using Feature Based Die-Map Clustering in the Semiconductor Manufacturing Process. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 8(2), 2014.

[19] Adam Rogers. "Samsung, Toshiba, and SanDisk lead Global NAND Flash Market", 2015. http://marketrealist.com/2016/01/samsung-toshiba-sandisk-lead-global-nand-flash-market/, accessed February 2016.

[20] Saul Rosa and Anton Vladimirov. Early defect identification of semiconductor processes using machine learning (Stanford University, Machine Learning CS229, Professor Dr. Andrew NG). December 16, 2011. http://cs229.stanford.edu/proj2011/RosaVladimirov-EarlyDefectIdentificationOfSemiconductorProcessesUsingMachineLearning.pdf, accessed October 2015.

[21] SanDisk. "SanDisk Corporation 2014 Annual Report", 2015. http://s21.q4cdn.com/620650385/files/doc_financials/annual/2014/SanDisk-Corporation-2014-Annual-Report-and-Proxy.pdf, accessed February 2016.

[22] SanDisk. "SanDisk Storage Solutions". Internal Company Presentation, 2015.

[23] SanDisk. "SanDisk Company History", 2016. https://www.sandisk.com/about/company/history, accessed February 2016.

[24] Tobias Sing. CRAN R-Project package ROCR, 2015-03-26. https://cran.r-project.org/web/packages/ROCR/ROCR.pdf, accessed August 2015.

[25] Jemmy Soenjaya, Wynne Hsu, Mong Li Lee, and Tachyang Lee. Mining wafer fabrication: framework and challenges. *Next Generation of Data-Mining Application, John Wiley & Sons, New York*, pages 17–40, 2005.

[26] Gian Antonio Susto, Simone Pampuri, Andrea Schirru, Guiseppe De Nicolao, Sean McLoone, and Alessandro Beghi. Automatic control and machine learning for semiconductor manufacturing: Review and challenges. In *Proceedings of the 10th European Workshop on Advanced Control and Diagnosis (ACD 2012)*, 2012.

[27] Tomio Tsuda, Shinji Inoue, Akihiro Kayahara, Shin-ichi Imai, Tomoya Tanaka, Naoaki Sato, and Satoshi Yasuda. Advanced Semiconductor Manufacturing Using Big Data. *IEEE Transactions on Semiconductor Manufacturing*, 28(3):229–235, 2015.

[28] Jeremy VanDerWal. CRAN R-Project package SDMTools, 2014-08-05. https://cran.r-project.org/web/packages/SDMTools/SDMTools.pdf, accessed August 2015.

[29] Lidong Wang and Cheryl Ann Alexander. Big Data in Design and Manufacturing Engineering. *American Journal of Engineering and Applied Sciences*, 8(2):223, 2015.

[30] Sholom M. Weiss, Amit Dhurandhar, and Robert J. Baseman. Improving quality control by early prediction of manufacturing outcomes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1258–1266. ACM, 2013.

[31] Hadley Wickham. CRAN R-Project package ggplot2, 2015-03-17. https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf, accessed August 2015.

[32] Toby Wolpe. "The rise of flash storage: Will it save every one of us?", 2015. http://www.zdnet.com/article/the-rise-of-flash-storage-will-it-save-every-one-of-us/, accessed February 2016.

[33] Lihui Wu, Jie Zhang, and Gong Zhang. A fuzzy neural network approach for die yield prediction of wafer fabrication line. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference*, volume 3, pages 198–202. IEEE, 2009.

[34] Shanghang Zhang, Xin Li, RD Blanton, Jose Machado da Silva, John M. Carulli, and Kenneth M. Butler. Bayesian model fusion: enabling test cost reduction of analog/RF circuits via wafer-level spatial variation modeling. *Test Conference (ITC), 2014 IEEE International*, pages 1–10, 2014.