Multidimensional Parametric Test Set Optimization of Wafer Probe Data for Predicting in Field Failures and Setting Tighter Test Limits

Dragoljub (Gagi) Drmanac,[1] Nik Sumikawa,[1] LeRoy Winemberg,[2] Li-C. Wang,[1] Magdy S. Abadir[2]
[1]University of California, Santa Barbara, [2]Freescale Semiconductor, Inc.

Abstract

This work proposes a wafer probe parametric test set optimization method for predicting dies which are likely to fail in the field based on known in-field or final test fails. Large volumes of wafer probe data across 5 lots and hundreds of parametric measurements are optimized to find test sets that help predict actually observed test escapes and final test failures. Simple rules are generated to explain how test limits can be tightened in wafer probe to prevent test escapes and final test fails with minimal overkill. The proposed method is evaluated on wafer probe data from a current automotive IC with near zero DPPM requirements resulting in improved test quality and reduced test cost.

1 Introduction

Parametric testing of the latest integrated circuit technology requires a delicate balance between test quality and test cost. Finding a proper balance can be extremely difficult especially when faced with large volumes of wafer probe data that is inherently noisy, expensive to capture, and statically varying. There is a constant demand for higher test quality, sometimes approaching nearly zero defective devices per million (DPPM), while requiring the lowest possible test cost. In practice, intrinsic and latent defects may result in final test fails or test escapes that must be dealt with during production. The conflict between product quality and test cost requires a shift from traditional one dimensional pass/fail screening to multidimensional data-driven methods capable of diagnosing and predicting failures [4].

In this paper we propose an early detection strategy for predicting both final test and in-field failures based on multidimensional test set optimization of parametric wafer probe data. Additionally, we show how some of these fails can be avoided by properly adjusting specific test limits without incurring significant overkill. This approach meets the requirements of improving test quality by preventing defective devices from reaching customers, while reducing test cost by early screening of borderline devices likely to fail burn-in. Therefore, avoiding unnecessary packaging cost while increasing burn-in and final test capacity.

Recently, there have been several papers proposing methods for reducing parametric test cost while improving quality [6][9][10]. These methods either select an essential subset of tests or directly predict failures based on statistical models constructed from test data. Often in these works, parametric tests are treated as a single measure of product quality where each test is analyzed individually from a single dimensional perspective. In contrast, this work takes a multidimensional perspective, where an essential combination of parametric tests are considered together when predicting failures.

Statistical outlier methods are used in [6] to detect burn in fails based on wafer probe data. Essential parametric tests are selected by correlating to known burn-in failures and outliers are identified by comparing location on wafer residuals for the selected tests. While this method shows practical promise, only a few wafers were analyzed from a one dimensional test perspective potentially missing key multidimensional interactions in failure data.

Parametric analog and RF test sets are optimized based on a defect density fault model in [10]. Open, short, and pinhole faults were simulated for analysis, which considerably limits the scope of possible failures. The proposed method was able to achieve zero DPPM for these specific faults. However, in practice, a robust parametric fault model does not exist and may be difficult to develop and validate.

Authors in [9] used several machine learning techniques to detect failing devices based on a subset of mostly functional and some parametric tests. The goal was to prevent test escapes and minimize overkills while reducing cost by removing non-essential analog tests. While the work employed a multidimensional test perspective, the proposed method was evaluated on fewer than one thousand dies.

In this work, we predict in-field and final test failures using outlier analysis and decision tree classification based on an optimized set of parametric tests. A two-class support vector machine (SVM) optimization algorithm is used to select essential wafer probe tests based on known in-field and final test failures. The optimized test set is then processed using one-class SVM multivariate outlier analysis, to rank dies based on their likelihood to fail in the field or

978-3-9810801-7-9/DATE11/

during burn-in. Additionally, classification and regression tree (CART) algorithms are used to explain failures, showing how to tighten a few key test limits to capture observed test escapes with minimal overkill.

The three key contributions of this work are (1) an optimization method for selecting key parametric tests that explain known failures, (2) a multivariate outlier analysis algorithm for detecting unseen failures using the set of optimal tests, and (3) a decision tree method showing how to tighten test limits.

The rest of the paper is organized as follows: Section 2 describes the process of selecting an essential test set which best describes observed failures. Section 3 explains how the results of test set optimization can be leveraged to detect failures as multivariate outliers. Section 4 discusses how to use decision trees to set better test limits. Section 5 summarizes the experiments and results, while Section 6 concludes the paper.

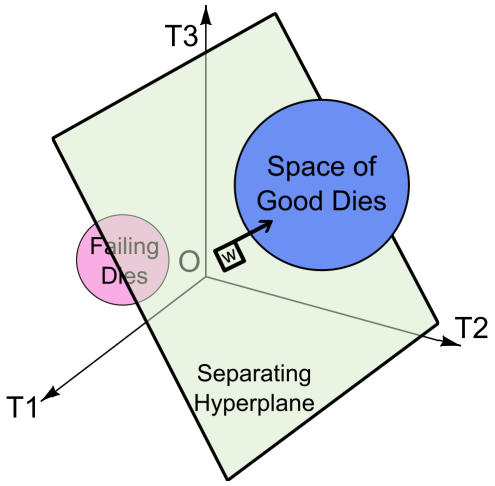

Figure 1. Hyperplane Separating Failing and Passing Dies in Three Test Dimensions

2 Learning From Known Failures

Large volumes of wafer probe data encode information about failing devices that can be leveraged to predict failures in future production steps potentially reducing burn-in cost, packaging cost, and expensive customer return diagnosis. Learning from known failing dies using wafer probe data can be performed by support vector machine (SVM) optimization algorithms. In this work, we use two-class and one-class ν-SVC algorithms described in [7] and [8], and a modified version of the open source LibSVM software package implemented by [3].

2.1 Parametric Test Set Optimization

Parametric test set optimization is performed using a two-class SVM predictive classifier that separates passing from failing dies based on labeled examples of known in-field or final test fails. A training sample is a vector of normalized wafer probe test measurements taken for each die. Each training sample is a pair $(\vec{s_i}, l_i)$, where $\vec{s_i} = (t_1, t_2, ..., t_n)$ contains n parametric test measurements $t_1 - t_n$, and l_i is a pass/fail classification label determined by final test binning or known in-field failures.

To find an optimal parametric test set, the ν-SVC soft margin classifier is used to separate passing and failing dies with maximum margin in a n dimensional wafer probe test space. Since the test space is large, possibly containing hundreds of test measurements, many separating hyperplanes exist. Optimization is used to find the best hyperplane that separates passing from failing dies with maximal margin using the following quadratic programming formulation [8].

$$Maximize \quad -\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j l_i l_j K(\vec{s_i},\vec{s_j})$$

$$Subject\ to \quad 0 \le \alpha_i \le \frac{1}{m},\ \sum_{i=1}^{m}\alpha_i l_i = 0,\ \sum_{i=1}^{m}\alpha_i \ge \nu.$$

Where $\alpha_{i,j}$ are Lagrangian multipliers, $\vec{s_{i,j}}$ are example dies, $l_{i,j} \in \{\pm 1\}$ are pass/fail training labels, ν is the fraction of accepted training errors, $K(\vec{s_i}, \vec{s_j})$ is the dot product kernel, and m is the number of samples [8]. In general, passing and failing dies may not be linearly separable so ν is used to control the fraction of misclassified samples allowed when establishing the classification boundary.

Figure 1 shows an idealized view of a hyperplane obtained after test set optimization in three test dimensions. In this simple example, for illustrative purposes, the hyperplane completely separates failing and passing dies. In general, a clear separation may not be possible depending on pass/fail device labels or the number of test dimensions. The key to multivariate test set optimization is that an optimal hyperplane can always be found that best separates passing and failing dies regardless of class overlap [7]. It was shown in [1] that the normal vector $\vec{w}$, which defines the linear hyperplane orientation, encodes the importance of each test when separating passing and failing dies.

An optimized test set can be extracted by selecting the top m tests as weighted by the components of the optimal separating hyperplane's normal vector $\vec{w}$ shown in Figure 1. The components of this vector describe how far the hyperplane has tilted in each test dimension to correctly classify failing devices. In Figure 1, tests T2 and T3 are weighted higher than test T1 since the weight vector points in the di-

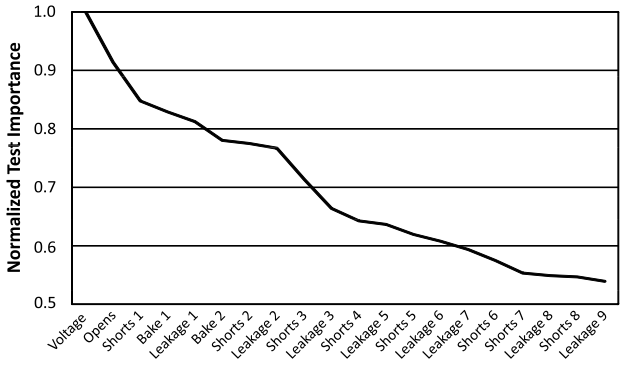

Figure 2. Top 20 Wafer Probe Tests Selected by Parametric Test Set Optimization

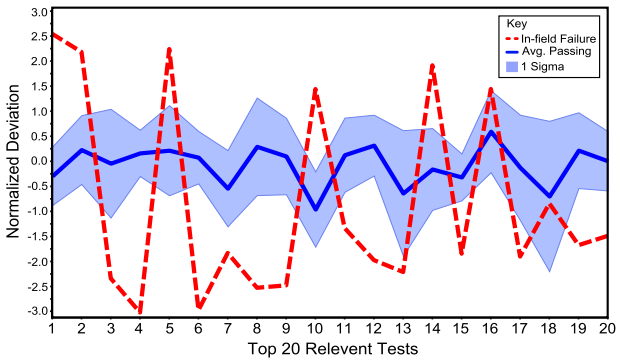

Figure 3. In-Field Failure Behaves as a Multivariate Outlier in Top 20 Relevant Wafer Probe Tests

rection of T2 and T3. This makes sense because passing and failing dies can clearly be separated by a linear combination of tests T2 and T3, making T1 less relevant. Another way to see this is that the hyperplane is almost perfectly parallel to test direction T1 showing that no change in test T1 will impact classification of the die. Thus, finding an optimized test set amounts to finding the best linear combination of tests that separates passing and failing dies. We can use this optimized test set to later predict in-field or final test failures and set better test limits.

An example test set optimization result based on one known in-field failure is shown in Figure 2. The x-axis depicts the top twenty most important tests and the y-axis shows their normalized relevance. Here we see that the top 20 tests cover about 50% of relevant information describing the failure mechanisms for this in-field fail. Note the similarities in selected test types, indicating the failure is likely related to problems with leakage and shorts.

3 Predicting In-Field and Burn-In Failures

In-field and final test failures are predicted using one-class SVM multivariate outlier analysis. An actual in-field failure is plotted in Figure 3. The x-axis shows the top 20 tests selected by test set optimization and the y-axis shows the normalized deviations for each test. The dashed line shows the in-field failure's wafer probe measurements and the solid line shows average measurements for nominal dies within one shaded standard deviation. From this multidimensional perspective, the in-field failure clearly behaves as a multivariate outlier in the 20 relevant tests selected by test set optimization.

In Figure 3, we clearly see how the in-field fail deviates from the normal trend established by good dies despite always being within 3 sigma test limits across all 20 measurements. Therefore, we show that test set optimization finds relevant tests to demonstrate the abnormal behavior of the in-field failure, from a multidimensional perspective, despite being quite normal for each test individually. This explains why the device was initially binned good. Abnormal multivariate behavior cannot be seen from a single dimensional test perspective.

Multivariate outlier analysis is required because it allows devices to be compared from multiple test perspectives simultaneously. This enables detection of subtly abnormal dies even though all individual test measurements may be well within nominal wafer probe limits. From a single dimensional perspective, the test escape in Figure 3 would never be labeled failing because it is always within three sigma test limits for all tests. However, multivariate analysis shows this device is actually borderline and abnormal in the top 20 essential tests found by test set optimization.

3.1 Multivariate Outlier Analysis

Outlier analysis is performed using one-class SVM on the top 20 tests found by test set optimization based on known final test or in-field fails. To analyze the wafer probe data each measurement is normalized to zero mean and unit variance, so different types of measurements are on comparable scales. Each die is represented as a vector of the top 20 wafer probe test $\vec{s_i} = (t_1, t_2, ..., t_{20})$. In this case, we have only selected 20 relevant tests out of the possible 730 parametric tests analyzed during test set optimization. In outlier analysis there are no training labels since the algorithm assumes all samples are of the same class and looks for abnormally behaving devices. The goal is to identify defective devices by maximally enclosing all good samples in a 20 dimensional hypersphere using the following quadratic programming optimization.

$$Minimize \ \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j K(\vec{s_i},\vec{s_j})$$

$$Subject\ to \ 0 \le \alpha_i \le \frac{1}{\nu m}, \ \sum_{i=1}^{m}\alpha_i = 1.$$

Where $\alpha_{i,j}$ are Lagrangian multipliers, $\vec{s_{i,j}}$ are training samples, ν is the percent of outliers to identify, and $K(\vec{s_i}, \vec{s_j})$ is the Gaussian kernel. After optimization, the outlier measure for each sample is computed using the function $g(\vec{x})$ shown below, where ρ is a constant offset.

$$g(\vec{x}) = \sum_{i=1}^{m} \alpha_i K(\vec{x}, \vec{s_i}) - \rho$$

In general, there will be outliers of various degrees, so it is important to perform an outlier ranking to examine the top outliers. $g(\vec{x})$ returns positive and negative numbers, where positive numbers correspond to normal samples and negative numbers correspond to outliers. Using $g(\vec{x})$, we select the top outliers as likely defective devices.

4 Setting Tighter Test Limits

During wafer probe testing, several hundred to a thousand measurements are taken to evaluate various device specifications. In some cases, physical measurements have no hard limits and are not directly tied to a spec. Instead, derived tests are computed based on measurements to determine if a device meets requirements. In such a scenario, test decisions are made based on computed values of physical measurements with wider or open limits. Using test set optimization it is possible to search the space of essential tests and find individual measurements whose limits can be tightened to capture final test or in-field failures with minimal overkill.

4.1 Decision Trees and Stumps

Decision tree classification is used to find a few key tests that best split passing dies from known in-field or final test fails. In decision tree learning, each node in a tree represents a limit for one test and each branch represents a splitting condition such as Leakage Current $> 4\sigma$, where σ is one standard deviation. Leaf nodes of the tree represent test decisions and contain the distribution of escapes and overkills. A path from the root to a leaf node represents a conjunction rule explaining a failure mechanism. The key to building a tree is finding one limit for each test that best splits passing and failing dies. Choosing the best split is accomplished using the information gain ratio heuristic introduced in [2].

Figure 5 shows examples of decision trees and stumps. Decisions trees contain multiple splitting condition as in (d) and (e), while decision stumps have only one splitting condition as shown in (a) (b) (c). Decision stumps can be used to set tighter test limits for a single specific test that will capture in-field or final test failures and avoid excessive overkill, whereas decision trees establish a multidimensional test decision requiring the evaluation of multiple test limits in succession before a pass/fail decision can be made.

5 Experiments and Results

We performed parametric test set optimization, multivariate outlier analysis, and decision tree learning on 5 lots of wafer probe data. Each selected lot was one of hundreds in production containing more than 10k passing devices, and one known in-field failure. The test suite consisted of 730 unique measurements from a current 130 nm automotive SOC design with near zero DPPM requirements. Additionally, final test binning information was used to label devices that pass wafer probe and fail final test for optimization. Experiments were performed using RapidMiner, an open source rapid prototyping tool for data mining [5].

Parametric test data was extracted from STDF files and normalized before analysis could take place. Extreme outliers greater than 6σ and missing or corrupted measurements were removed. The wafer probe test suit used in this work consisted of various types of open, short, leakage, Idd, and memory tests. Within each lot all measurements were normalized to zero mean and unit variance to ensure comparable scales between different test types.

5.1 Predicting In-Field Failures

To evaluate the proposed method we performed multidimensional test set optimization based on a single in-field failure and used the optimized test set to predict five simulated as well as other actually observed in-field fails. Simulated failures were generated by randomly adding up to $\pm 15\%$ noise to an observed failure.

Test set optimization was performed on each lot of parametric wafer probe data containing roughly 10k dies with 730 test measurements each. Only dies passing wafer probe testing were considered for analysis. The known test escape was labeled as a failing example and the rest of the dies were labeled as passing. Test set optimization was performed using two-class SVM. The top 20 most relevant parametric tests were selected based on the weight vector describing the orientation of the maximal margin hyperplane separating the known test escape from the rest of the good dies.

After test set optimization, one-class SVM outlier analysis was used to predict other samples likely to be test escapes. Only the 20 most relevant tests were used during outlier analysis. Table I shows the results when predicting known and simulated test escapes as multivariate outliers across 5 lots. The columns show the lot ID, number of dies in each lot, outlier rank of the actual in-field failure, and the worst rank of the simulated test escapes. The next three columns show the number of final test fails and overkills captured within the top 10, 20, and 30 outliers respectively.

Table I shows that after learning from known test escapes, it is possible to predict similar in-field fails as well as capture additional final test failures with high accuracy. Across all 5 lots, the actual test escape was detected as the

Lot	Num. Dies	TE Rank	SE Rank	Top 10		Top 20		Top 30	
				F	K	F	K	F	K
1	8354	1	2	2	7	3	16	4	25
2	9389	1	1	1	8	2	17	3	26
3	11998	1	1	2	7	2	17	3	26
4	11049	1	1	0	9	2	17	2	27
5	9519	1	1	1	8	1	18	1	28

Test Escape (TE), Simulated Escapes (SE), Final Test Fail (F), Overkill (K)

Table I: Predicting Test Escapes As Multivariate Outliers

highest ranked outlier, and the simulated test escapes were at worst ranked as the number two outlier. For all lots, overkills were kept to a minimum. The top 10 outliers captured all test escapes, this equates to roughly 0.1% overkill. For products with very low DPPM requirements catching a test escape with only 0.1% overkill was deemed an acceptable tradeoff between test cost and test quality.

Table II shows the performance of predicting one test-escape based on test set optimized on another. For example, the test escape in lot 1 can be used to select an optimized test set to catch the test escape in lot 5. Similarly, test set optimization based on the escape in lot 2 can predict the test escape in lot 4. In both cases, overkills were kept below 2% showing that it is possible to learn from one test escape to predict another.

Train Lot	Predict Lot	TE Rank	% Overkill	FT Fails	Relevant Tests
1	5	43	0.47	4	Leakage
2	4	167	1.64	13	Memory

Table II: Predicting Test Escapes Between Lots

The results in Table II indicate that there exist similarities between some test escapes and that an optimized test set can be used to detect other similarly failing dies. For example, borderline leakage tests may explain the similarity between test escapes from lots 1 and 5, while common failures in memory tests may explain the ability to predict test escapes between lots 2 and 4. This result is encouraging, because it gives test engineers a mechanism to learn from past in-field fails and help prevent future escapes in different lots without having to explicitly diagnose an exact failure mechanism.

5.2 Predicting Test Escapes from Burn-In Fails

To determine if it was possible to predict test escapes from post-burn-in failures, we performed test set optimization based on known final test failures. Only dies passing wafer probe test were considered. Dies passing wafer probe but failing final test were labeled as failing examples and the rest (including the test escape) were labeled as passing. Test set optimization found relevant tests by maximally separating final test fails from all good dies in the space of wafer probe measurements. The top 20 tests were selected for subsequent multivariate outlier analysis.

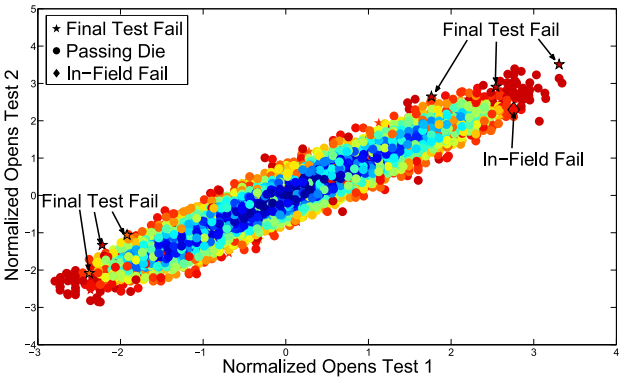

Figure 4. Top Test Dimensions Showing In-Field and Final Test Fails as Outliers in Opens Tests

Lot	TE Rank	% Overkill	FT Fails	Relevant Tests
1	29	0.29	5	Leakage
2	65	0.64	5	Opens/Memory
3	61	0.51	1	Leakage/Shorts
4	150	1.27	10	Memory
5	41	0.37	6	Leakage

Table III: Predicting Test Escapes from Final Test Fails

One-class SVM was used to predict test escapes based on the optimized test set learned from final test fails. The goal was to detect actually observed test escapes by learning from marginal devices that failed final test after burn-in. Table III shows the results of predicting test escapes when learning from final test fails. Columns show the lot ID, test escape rank, percent overkill and the number of final test fails captured in the top ranked outliers.

The results show that across most lots, learning an optimized test set from known final test fails can help predict unforeseen test escapes as well as some final test failures. Since test set optimization was not directly preformed on the known test escapes we expect slightly more overkill. Fortunately, overkill was shown to fall below 1% for most lots which justifies the early prediction of test escapes based on learning from known final test failures. For products requiring near zero DPPM, $\leq 1\%$ overkill may be justified by cost savings in failure analysis and company reputation.

Figure 4 shows a scatter plot of two top tests found by test set optimization of devices in lot 2. In this lot, many relevant tests were opens measurements, so both the x-axis and the y-axis show normalized opens test results. The plot points out several final test fails as well as the single observed in-field fail. Point color corresponds to the outlier rank, with red being more outlying and blue being more normal. Already, in only two relevant test dimensions, we

see final test and in-field fails appearing at the edges of the distribution behaving as outliers. In higher dimensions failing dies are even more apparent outliers.

5.3 Setting Better Test Limits

Decision tree classification is applied to the optimized test sets learned from observed in-field fails. An information gain ratio heuristic was used to search the optimized subset of tests for tight limits that can capture the observed test escape with minimal overkill. The minimal information gain ratio was set to .01 with a maximal tree depth of 2. The aim was to find simple trees that can explain the in-field fail in the space of relevant tests. In some cases, a decision tree with only one node was found, which directly shows how to tighten a specific test limit to avoid an observed test escape.

The results of decision tree classification are summarized in Figure 5. Each tree contains nodes which are specific tests, and each branch shows an applied test limit normalized by one standard deviation. Leaves contain a test decision as well as the number of passing/failing dies which fall in the leaf. For example, Figure 5 (b) shows a decision stump that reads: "if a memory test measurement is less than -3.74 standard deviations the die is a test-escape, otherwise it is a passing die". In the fail leaf we see 3/1 indicating that 3 passing dies and one test escape are detected by the -3.74 sigma test limit applied to a memory test.

From the decision trees we see that specific test limits can be tightened to detect observed in-field fails for lots 1, 2, and 3. In each case overkill is kept below 0.4%. For a product with near zero DPPM requirements loosing 0.4% yield to capture test escapes may be justified, especially since tightening test limits can be easily performed without any change to the existing test flow.

Lots 4 and 5 have no single test split that captures in-field fails with minimal overkill, but a two dimensional test split captures the test escape with zero overkill. For example, Figure 5 (e) reads: "if a voltage test is less than -1.35 sigma and at the same time an opens test is less than -1.51 sigma the device is a test escape, otherwise the device is good". In this case, a multidimensional test limit may be implemented in the test flow to check if other devices behave similarly. From our experiments the test escape is captured with zero overkill by setting a two dimensional test limit. These results show real promise for the inclusion of multi dimensional testing in actual test flows.

6 Conclusion

This work proposed a wafer probe test set optimization method for selecting essential parametric tests that can be used to predict in-field or final test fails and set better test limits early in production. Multivariate outlier analysis was applied to the 20 most relevant tests selected by learning

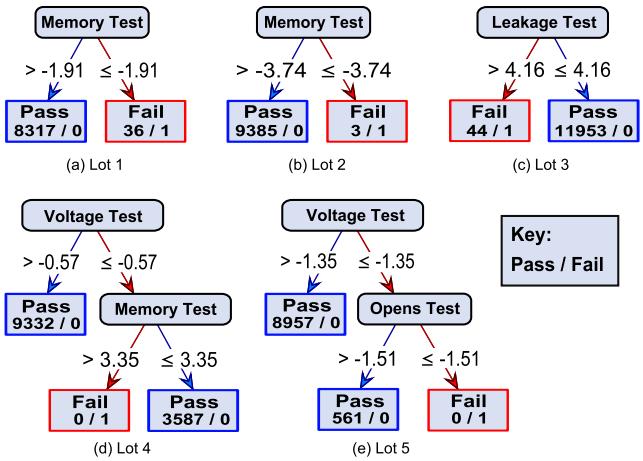

Figure 5. Setting Test Limits that Capture In-Field Failures Using Decision Trees

from known final test or in-field fails and used to predict other test escapes. Optimized test sets were analyzed using decision trees to set tighter limits and capture known in-field failures without incurring significant overkill. A case study across 5 lots of wafer probe data proved the method's ability to effectively learn from observed failures and detect problems early in production.

References

[1] J. Brank, M. Grobelnik, N. Milić-Frayling, and D. Mladenić. Feature Selection Using Linear Support Vector Machines. In *Microsoft Research Technical Report*, 2002.

[2] L. Breiman, J. Friedman, C. J. Stone, and R. Olshen. *Classification and Regression Trees*. Wadwosrth, 1984.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] W. R. Daasch, C. G. Shirley, and A. Nahar. Statistics in Semiconductor Test: Going beyond Yield. In *ITC*, 2009.

[5] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid Prototyping for Complex Data Mining Tasks. In *SIGKDD*, 2006.

[6] A. Nahar, K. M. Butler, J. M. C. Jr., and C. Weinberger. Quality Improvement and Cost Reduction Using Statistical Outlier Methods. In *ICCAD*, 2009.

[7] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the Support of a High-Dimensional Distribution. In *Neural Computation*, 2001.

[8] B. Scholkopf, A. Smola, R. Williamson, and P. L. Bartlett. New support vector algorithms. In *Neural Computation*, 2000.

[9] H.-G. D. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris. Non-RF To RF Test Correlation Using Learning Machines: A Case Study. In *VTS*, 2007.

[10] E. Yilmaz and S. Ozev. Defect-Based Test Optimization for Analog/RF Circuits for Near-Zero DPPM Applications. In *ICCAD*, 2009.