Understanding Customer Returns From A Test Perspective

Nik Sumikawa, Dragoljub (Gagi) Drmanac,
Li-C. Wang
University of California, Santa Barbara

LeRoy Winemberg,
Magdy S. Abadir
Freescale Semiconductor, Inc

Abstract

Customer returns are defective parts that pass all functional and parametric tests, but fail in the field. To prevent customer returns, this paper analyzes wafer probe test data and tries to understand what it takes to screen them out during testing. Because these parts pass all tests, analyzing their signatures based on the original test perspective does not make sense. In this work, we search for a novel test perspective where the test signatures from parametric measurements can be used to separate the returned parts from the rest of population. Our study shows that in order to effectively screen customer returns during wafer test, a multivariate screening methodology is desired. This study is based on analyzing over 1000 parametric wafer probe tests and dies from seven lots, each lot containing one returned part. We demonstrate that analyzing customer returns from a multivariate test perspective leads to robust and conservative results.

1 Introduction

In a market that requires high quality products with near zero defective parts per million (DPPM), the effects of a test escape can be significant in terms of debug and diagnosis costs. More importantly, an excessive number of test escapes will damage a company's reputation and can lead to missed business opportunities. For this reason, quality is extremely important and often, it outweighs the costs of overkills. Examples are automotive products where a test escape can have significant consequences.

To reduce DPPM to near zero, the logical first step is to acquire a better understanding of customer returns. Traditionally, this is achieved by diagnosis to a root cause. This work does not follow a root cause analysis approach. Instead, our objective is to understand customer returns from a behavioral point of view where its behavior is reflected in the values of parametric measurements. We call this a test perspective. Instead of focusing on the root cause, we try to understand the behavior of customer returns in order to develop a new test strategy to screen them in the future.

It should be noted that there have been several proposed methods for improving the quality of parametric testing. Generally, these methods employ test selection and/or some statistical learning techniques to detect potential failures. For example, the authors in [4] used test selection and outlier analysis to predict burn-in failures. First, a smaller set of tests was derived from known failures which was then used to compare dies residing within the same wafer residuals in order to predict other defective devices. This analysis was based on single test measurements to ensure simplicity and applicability in practice.

As another example, the work in [7] analyzed a dataset of functional and parametric results for screening RF devices. A subset of important tests was extracted and machine learning algorithms were used to identify defective devices. This screening strategy was effective and the work was a good comparison of various algorithms.

In this work, the problem context is different. First, the product is a SoC for the automotive market where a large portion of the design is memory and analog. Second, we analyzed seven customer returns that passed a comprehensive testing process where the quality requirements are very high. In other words, the DPPM was already close to zero, but not exactly zero. Finally, the seven customer returns belong to different lots and we had access to over 1000 parametric wafer probe test measurements for each die across the seven lots.

It is important to clarify the problem context for two reasons. First, a different context may imply a different problem. For example, the challenges in analyzing a situation with 250 DPPM with the goal of pushing to 100 DPPM could be very different from the challenges of pushing from 50 DPPM to 10 DPPM. Similarly, analyzing a SoC could be different from analyzing an analog device. Second, when applying the findings in this work, one should carefully consider the context for their respective application. For example, in a different DPPM range, a customer return can behave differently than the ones studied in this work. Hence, one should be cautious and check the assumptions in this work before applying our methods.

Instead of jumping into the development of a screening methodology, we first ask the key question: "How did we miss the seven customer returns?" After all, the testing was

978-1-61284-656-9/11/$26.00 ©2011 IEEE

very comprehensive. This key question leads to the following three questions:

- Is it because we still lack the right test(s)?
- Is it because we did not set the test limits correctly? Were some of the limits set too conservatively?
- Is it because we did not look at the tests from the correct perspective? If so, what is the correct perspective?

Section 2 investigates these three questions. Our conclusion will be that we do not need more test(s), nor test limit adjustments, as long as the screening takes a multivariate perspective, i.e. making decisions based a collection of tests. Section 3 describes the process of selecting relevant tests, followed by section 4 that explains the role of test selection in multivariate outlier analysis. Section 5 discusses experimental results to show how such an outlier model works. Section 6 concludes.

2 The Three Questions

Questions 2 and 3 should be analyzed first because yes to either question means additional test(s) are not needed. In the following, we discuss these questions based on seven lots of data each consisting of ∼12,000 dies.

Prior to analysis, the test data was cleaned by removing all dies that failed during wafer probe testing and those with missing measurements. The non-parametric tests were also removed. Each measurement value was normalized to zero mean and unit variance, which ensures compatibility between different test types. The resulting parametric test set contains various types of measurements including opens, shorts, leakages, Idd and memory tests.

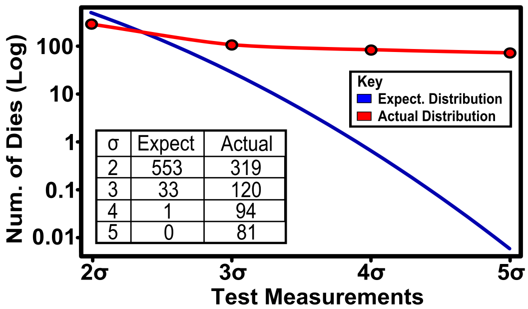

Figure 1. Gaussian is not a good assumption at the tail

2.1 How effective is using a kσ rule?

Due to a lack of knowledge, the parametric test limits are often set using the empirical kσ rule, where an example may be k = 3, 6, etc. Figure 1 shows the behavior of a particular parametric measurement for one lot of dies. Assuming a Normal distribution and setting a 4σ limit, we would expect to see 1 die beyond this limit. However, we observed 94 dies. Keep in mind that all of these dies passed wafer testing (not necessarily after final testing), including the one customer return. Hence, a 4σ test limit is too aggressive and it would result in the overkill of 94 good dies.

For this reason, test limits are usually set with a much larger σ to minimize overkill. Hence, the customer return may have been missed due to this conservatism.

Customer Return	applied to all tests	applied to only the specific test	specific test info σ	specific test info Type
1	43.19%	0.71%	-3.02	Memory
2	42.41%	0.03%	3.45	Memory
3	13.02%	0.20%	8.08	Voltage
4	3.20%	0.20%	19.43	Leakage
5	5.33%	0.48%	4.80	Leakage
6	19.90%	0.23%	6.23	Leakage
7	27.26%	1.02%	4.01	Current

σ found with the best test to minimize overkill and screen out the return

Table 1. Overkill % based on applying the σ rule

Table 1 shows our hypothesis is not true. For each returned part, we found the σ value for each parametric test. For example, 5σ means the measured value using the test is at the 5σ point of the distribution whose mean and σ are calculated based on all dies in the lot. We then found the specific test whose σ value is the largest and used this value as a kσ rule. For example, customer return #1 in Table 1 has a value of -3.02σ corresponding to a memory test.

Suppose we apply the -3.02σ rule to the specified test only in order to screen the dies (all dies with measured values whose absolute value $\geq$ 3.02σ are screened out), we would have an overkill of 0.71% in the lot, which is ∼80 dies. This overkill is shown in the third column of Table 1.

This scenario assumes that we know which test is the best to use for the kσ rule. Suppose we do not know this and we applied the same kσ rule to all tests. In this case, the second column of the table show the overkill % for each lot containing the customer return.

Table 1 shows that with a kσ rule, we cannot screen out a customer return without incurring significant overkill. Based on column 3, it is not desirable to capture customer returns by adjusting test limits. Using part #7 as an example; even in the best case we would have screened out more than 110 good dies.

2.2 Multivariate test perspective

In Table 1, we are examining the tests one at a time. What if we examine two or three tests collectively? Figure 2 shows such results based on customer return #4. Interestingly, the returned part looks more like an outlier when analyzing tests collectively, i.e. it's easier to separate from the rest of the dies.

Figure 3 shows that a hyperplane can separate customer return #5 from all good dies in three test dimensions. Based on these two figures, a customer return can be screened out with minimal overkill when examining the data using the right combination of tests. The big question is: From more than 1000 tests, which tests should we use?

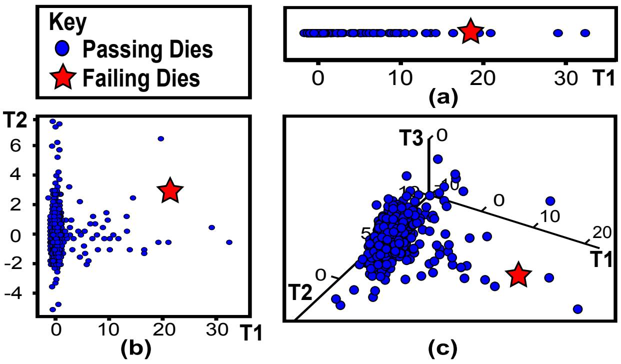

Figure 2. **Moving to multivariate test perspectives**

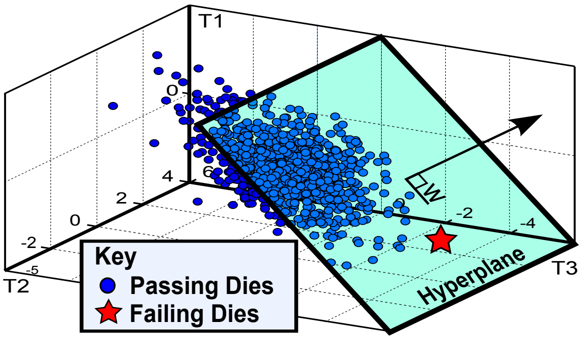

Figure 3. **Hyperplane Separating the Return from Other Dies in 3 Test Dimensions**

3 Learning the Relevant Tests

Based on the analysis above, we need to know the relevant tests in order to capture the customer return. In addition, it seems that the more tests we use, the more likely we are to be successful. For example, two tests are sufficient for return #5 while three tests are enough for return #4. Based on Table 1, we may conjecture that we will need even more tests for customer return #7.

The analysis also suggests a screening methodology to detect dies with similar behavior as the customer return. This method is based on the ability to learn a set of relevant tests from a returned device and show that the customer return can be identified as an outlier among all other dies in the lot. Using the same set of tests, outlier analysis can be performed on dies from future lots in order to identify potential customer returns that behave similar to the one used for learning.

To realize such a methodology, the first step is to have a learning method that can learn from the test data and select the relevant tests. The requirement for such a method depends on another question: Do we need to find a specific subset of tests relevant to the returned device for the outlier model to work effectively? In other words, how precise should the test selection process be? If we need a very precise set of tests; it can be challenging to develop such a learning method.

3.1 SVM and Chi Square

Given test data for the customer return and ∼12K dies, our goal is to rank the importance of the ∼1000 tests based on their ability to differentiate the returned part. This can also be thought as the following problem: Given two classes of samples and a set of features that describes the samples, rank the importance of the features in terms of their contribution to separate the two classes. This is commonly known as a feature ranking problem.

In the context of timing analysis, authors in [8] apply feature ranking to rank cells and nets by their contributions to path timing. This ranking is based on the linear Support Vector Machine (SVM), which is a binary classification approach described in [8]. Using the C-Support Vector Classification (C-SVC) algorithm with a dot-product kernel (i.e. a linear kernel) [5], we find an optimal hyperplane in n dimensions that best separates the two classes of samples. In our context, n is the number of tests examined and the samples are the dies. For example, Figure 3 shows a hyperplane in 3 dimensions. This hyperplane can be written as $f(T1, T2, T3) = w_1T1 + w_2T_2 + w_3T_3 + b$ where b is the constant defining the location of the hyperplane, i.e. where it intercepts the $T2$-$T3$ plane.

For a linear model, the normal weight vector $\vec{W}$ of the hyperplane encodes the importance of each test [10]. Similarly, the components w_1, w_2, w_3 of the weight vector can be thought as the importance of tests $T1, T2, T3$, respectively. These components describe how much the hyperplane is tilted in the direction of the test in order to correctly classify the customer return. In Figure 3, tests $T1$ and $T3$ are more important than $T2$ because the weight vector is pointed toward test $T1$ and $T3$. Test $T2$ is irrelevant since the hyperplane is almost parallel to the $T2$ axis [10].

Another common method for ranking features is the Chi-Square method [2]. Chi-Square is an algorithm that does not build a binary classification model as does a linear SVM. Instead, it calculates the importance of each test as a chi-square statistic. This statistic tries to calculates the amount of separation between two classes of dies using a single test, i.e. measuring its separation power. Note that this separation power is measured based on each test individually. Hence, this method ignores correlation among tests.

3.2 Test Selection

SVM and Chi-Square can output the test importance for each customer return. Given such a ranking, one still needs to select a subset of tests. For example, Figures 2 and 3 show how two and three tests can separate customer returns. These tests were manually selected based on the outlying behavior of the return. When using SVM or Chi-Square to select tests automatically, we can think of various questions:

- Will the top two or three tests, ranked by either algorithm, match our manual selection?

- If not, how do we determine the size of the test set so the desired tests are included?
- Do we need to have specific subset of tests for outlier analysis to work? If not, how many “irrelevant tests” can we include before outlier analysis breaks down? i.e. how flexible must test selection be for outlier analysis to work?
- Do the SVM and Chi-Square rankings agree?

The following discussion is centered around these questions and will demonstrate several interesting points based on our data. First, results from SVM and Chi-Square usually do not agree. If we focus on the top two or three tests, their rankings do not agree with our manual ranking. At first, this was seen as a critical barrier but we found that for outlier analysis to work (i.e. to identify a customer return as a top outlier), it does not require the use of specific tests. In fact, there is a high degree of flexibility in test selection, which is enabled by the outlier analysis. This means that we do not have to worry about selecting an exact subset of tests. Instead, we can include many irrelevant tests. This is an important property to note because it enables the development of a practical methodology that can learn from returned parts and screen out similar dies in the future.

If there is a high degree of flexibility when selecting tests, why not include all tests? Later, we will show that this flexibility is bounded. Hence, if we select a test set that is too large, outlier analysis will lose its effectiveness.

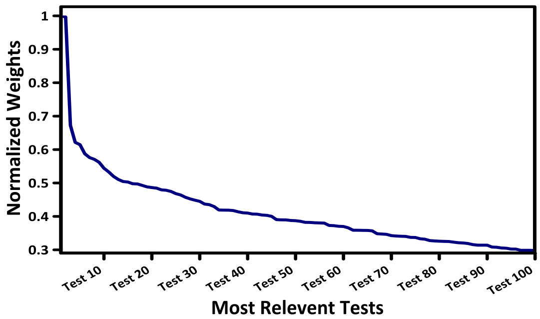

Figure 4. Test Importance for the Top 100 Tests

To demonstrate the output of the test ranking algorithm, Figure 4 shows the normalized importance of the top 100 tests using SVM ranking for customer return #1. From the weight curve in Figure 4, we can see that the importance starts to level out after the first 10 tests.

The behavior of customer return #1 is shown in Figure 5, where the tests are shown on x-axis. The tests are ordered by their importance, which is determined by the Chi-Square ranking this time. The measured values for each test are normalized by the variance. The behavior of the customer return is shown (red). The average measured value (mean) of each test, across all dies in the same lot, is shown (dark blue). One standard deviation (one σ) on either side of the mean is shown (light blue) for each test. A clear trend can be observed where the customer return deviates further from

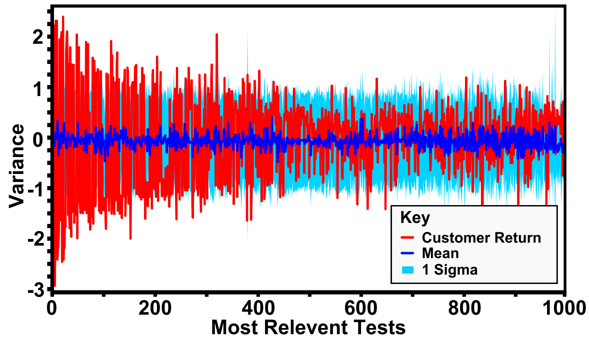

Figure 5. Effects of Diminishing Test Importance

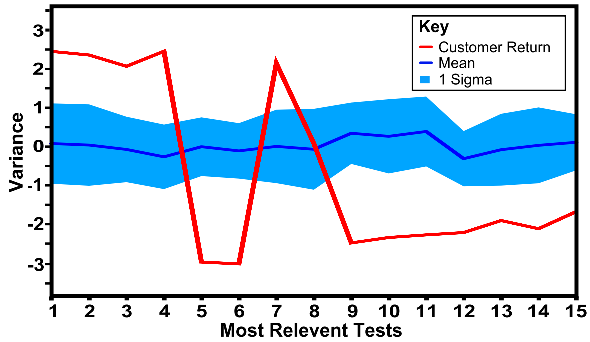

Figure 6. Return #1 as an Outlier on top 15 Tests

the mean in the higher ranked tests (on the left). Also, the measurements for the customer return reside within 3.02 σ limit as was also shown in Table 1.

Figure 6 zooms in on the top 15 tests. The customer return curve (red) is clearly different from the expected trend, which is shown as the mean$\pm 1\sigma$ band (blue). From this figure, the outlying behavior of the returned part is easily seen. Since the customer return's measurements are within $\pm 3\sigma$, this outlying behavior is only seen when examining the 15 tests collectively. Hence, this is a multivariate outlier.

4 Multivariate Outlier Analysis

It is important to note the following properties. If we present Figure 6 by removing the tests 11-15, it does not alter our ability to declare the customer return as an outlier. On the other hand, if we consider all tests in Figure 5 together, it is not clear if the customer return is an outlier. These two figures hint at an interesting property. There is a certain degree of flexibility in test selection that allows a customer return to be identified as an outlier. In the following section, we will illustrate this property further.

In this work, multivariate outlier analysis is performed using the one-class SVM algorithm on a subset of relevant tests. We use the one-class ν-SVM algorithm [6] with a Gaussian kernel [9] and a modified version of the open source LibSVM software package [1]. All experiments were performed under the software framework RapidMiner [3]. Here, we do not intend to describe the details of the outlier analysis algorithm. Instead, we are interested in studying the effects of test selection on outlier analysis.

To study this impact, we perform the following experiments. For each customer return, we apply a ranking method to rank the tests. From this ranking, we select the top k tests and create a dataset D_k. Each test in D_k contains the measurements for all the dies within the lot. Outlier analysis is performed on D_k using the one-class SVM algorithm. The results of outlier analysis is a ranking of the dies. Using these results, we identify the rank R_k of the customer return. In the best case we would have $R_k = 1$ and in the worst case we would have $R_k \approx 12{,}000$. In general, a $R_k \leq 20$ is considered very good.

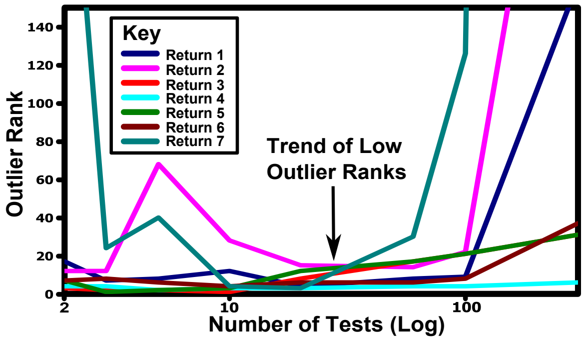

Figure 7. **Using Chi-Squared Test Selection**

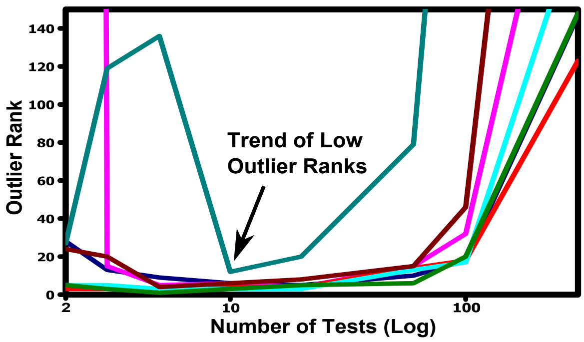

Figure 8. **Using SVM Test Selection**

Figure 7 shows the R_k (y-axis) for $k = 2 \ldots 300$ tests (x-axis) where the x-axis is in log scale. In this image, the results are based on Chi-Square test ranking and each customer return corresponds to a curve. In general, the trend says that if we use too few or too many tests, the results are not good. On the other hand, there is not much difference between using 10 to 50 tests.

For returns #1, 3, 4, 5 and 6, using a few tests is fine and using more (up to 50) does not hurt. This is not true for return #7, which requires using more than 10 tests. From Figure 7 we see that return #7 is a high-dimensional outlier ($\geq$ 10 test dimensions). Figure 8 shows similar results based on SVM test ranking. It can be clearly seen that SVM results do not exactly agree with Chi Square results, but a similar trend exists.

Customer Return	1	2	3	4	5	6	7
SVM; $k, R_k =$	20,5	20,4	3,3	10,3	5,1	5,4	10,12
Chi-Square; $k, R_k =$	20,4	3,12	10,1	5,2	3,1	10,4	20,3

Table 2: k **that gives the smallest** R_k **(k, R_k) in Figures 7 and 8**

Customer Return	1	2	3	4	5	6	7
SVM; $R_k =$	6	6	4	3	3	6	12
Chi-Square; $R_k =$	12	28	1	3	3	4	4

Table 3: R_k **Based on top 20 tests in Figures 7 and 8**

Table 2 shows the k value that gives the best R_k in Figures 7 and 8. For example, using SVM ranking on customer return #1, with $k = 20$, the returned part is identified as the 5th outlier ($R_k = 5$) according to the outlier analysis ("20,5"). To achieve the best result for different returns, we would need different numbers of tests. The best R_k is usually small except for return #7 using SVM and return #2 using Chi Square, where both are ranked $R_k = 12$.

Table 3 shows results (R_k) when selecting the top 20 tests (fix $k = 20$) for all returns. These results are similar to the best R_k presented in Table 2. This method further demonstrates the flexibility of the test selection because a small variation on the size of the test set does not have a significant impact on outlier analysis.

5 Screening Potential Returns

Suppose we learn the 20 most relevant tests for a customer return and verify that outlier analysis can rank the returned part as a top outlier. Suppose we take the 20 tests and perform outlier analysis on another lot. Can we identify dies that behave similar to the customer return?

This question can be studied from two perspectives: How similar is a die's behavior to the customer return and how much variability exists from one lot to another. In the following experiments, we report various results using customer return #1 and SVM test ranking.

In the first experiment, we take all of the test measurements from the customer return and inject $i\sigma$ noise on each value in order to make a "simulated customer return." This simulated return is put back into the dataset and we perform outlier analysis on the new dataset to see how the simulated return is ranked. Table 4 shows results for $i = 0, 0.5, 1.0$ and 1.5. For $i = 0$, the simulated return is the original customer return. The experiment is iterated 100 times thus simulating 100 returns.

In Table 4, "Rank" is the average R_k across 100 simulated returns. "E" is the number of simulated returns whose ranks exceed 50, i.e. assuming top 50 outliers are screened out, these dies would become test escapes. It is interesting to see (the "Rank" column) that using 2 tests does not tolerate the noise injected on the simulated returns as the average R_k is large. Using 2 tests can capture some simulated returns, but many (23-35) have a $R_k > 50$ and thus escapes detection. This shows that using 2 tests is not robust. This is still more effective than using 300 tests where outlier analysis is neither robust nor effective in capturing simulated returns.

Noise	2 Tests		10 Tests		100 Tests		300 Tests	
	Rank	E	Rank	E	Rank	E	Rank	E
0.0 σ	13.0	0	5.0	0	18.0	0	147	100
0.5 σ	294.8	23	6.0	0	16.7	0	415	100
1.0 σ	2392.3	35	5.6	0	14.7	0	347	100
1.5 σ	2545.6	32	4.7	0	10.9	0	205	88

Rank: Average R_k over 100 simulated returns, E: Number of Escapes

Table 4: Simulating Failures via Noise Injection

If we use 10 to 100 tests, the average R_k of the simulated returns does not change much with respect to different amounts of injected noise. Based on the results in Table 4, if we want a robust solution that detects potential returns whose behavior is similar to the known customer return, we cannot use too few or too many tests.

In the second experiment, we randomly select one good die and change its test measurements for only the top 2 tests based on SVM ranking. In particular, we replace the measured values of this good die with values similar to the customer returns. This die is added back to the dataset and outlier analysis is performed using various numbers of tests. Since the altered die is mostly good and behaves nominally for all but 2 tests, we expect it to have a poor rank and it should not be screened. The experiment was iterated 100 times and the results are shown in Table 5. According to this table, when using 2 or 3 tests, the altered good die would be classified as a top outlier and it would be screened. When more tests are used, the rank increases and fewer altered good dies are screened (20-33). Hence, using more tests ensures conservatism by capturing dies that are most similar to the customer return, which is a desired property to avoid overkill.

2 Tests		3 Tests		5 Tests		10 Tests	
Rank	S	Rank	S	Rank	S	Rank	S
1	100	1	100	570	33	630	20

Rank: Avg. Rank of Altered Good Die, S: # of Altered Good Dies Screened

Table 5: Conservatism Study in Outlier Analysis

Table 6 repeats the experiments shown in Table 4 with additional noise injected on the good dies. We altered the good dies by injecting 2% random noise on all of the test measurements. As it can be seen, the results in Table 6 are similar to the results in Table 4. This shows that small random variability across lots does not impact the effectiveness of outlier analysis as long as we do not use too few tests.

Noise	2 Tests		10 Tests		100 Tests		300 Tests	
	Rank	E	Rank	E	Rank	E	Rank	E
0.5 σ	429.6	25	5.1	0	17.7	0	407	100
1.0 σ	1794.2	35	5.1	0	17.1	0	352	100
1.5 σ	2594.6	37	3.2	0	15.3	0	211	88

Rank: Average R_k over 100 simulated returns, E: Number of Escapes

Table 6: Noise Injection on All Good Dies

In the last experiment, we take the top 50 tests learned from one lot and perform outlier analysis on another lot. Table 7 shows the results. If we learn from the returned part in lot 1, outlier analysis on lot 5 is able to classify its customer return as the 51st outlier. If we can tolerate 0.4% overkill, we can capture this customer return. In another case, we learn from the return in lot 2 and we can screen out the customer return in lot 4 if we are willing to tolerate 0.7% overkill. It is interesting to note that among the top outliers in lots 5 and 4, there were 4 and 10 dies actually failed final test, respectively.

Train Lot	Predict Lot	Rank	Overlap of top 50 tests
1	5	51	12
2	4	100	11

Table 7: Cross-lot Fortuitous Prediction

If we examine the top 50 tests learned from these four lots individually, lot 1 and lot 5 share 12 tests and lot 2 and lot 4 share 11 tests. This sharing may be used to explain the cross-lot fortuitous customer return detection.

6 Conclusion

In this paper, we study how to learn from existing customer returns and how to develop a methodology to screen other potential customer returns. Findings, based on studying seven customer returns from seven lots of data and more than 1000 parametric tests, are: (1) Customer returns can be effectively screened by multivariate outlier analysis. (2) To perform such an analysis on a lot of dies, we must first determine which relevant tests to use. (3) Relevant tests can be learned from existing customer returns. (4) The selection of relevant tests is not strict and has a degree of flexibility. In fact, using too few tests is not robust. When using at least 10 tests, outlier analysis becomes much more robust and using up to 100 tests can still be effective. (5) Using more tests, i.e. performing a high-dimensional outlier analysis, ensures both robustness and conservatism in capturing potentially defective parts whose behavior is similar to the returns we have learned from.

References

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

[2] G. Forman. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. In *Journal of Machine Learning*, 2003.

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

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

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

[6] B. Scholkopf, et al. Estimating the Support of a High-Dimensional Distribution. In *Neural Computation*, 2001.

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

[8] L. Wang, P. Bastani, and M. Abadir. Design-Silicon Timing Correlation - A Data Mining Perspective. section 4.2, Figure 8, *DAC*, 2007, pp. 384-389.

[9] S. Wu, D. Drmanac, and L. Wang. A Study of Outlier Analysis Techniques for Delay Testing. In *ITC*, 2008.

[10] D. Drmanac, N. Sumikawa, L. Winemberg, L. Wang, and M. Abadir. Parametric Test Set Optimization of Wafer Probe Data for Predicting in Field Failures and Setting Tighter Test Limit. In *DATE*, 2011.