Screening Customer Returns With Multivariate Test Analysis

Nik Sumikawa, Jeff Tikkanen and Li-C. Wang
University of California, Santa Barbara

LeRoy Winemberg and Magdy S. Abadir
Freescale Semiconductor, Inc

Abstract

This work studies the potential of capturing customer returns with models constructed based on multivariate analysis of parametric wafer sort test measurements. In such an analysis, subsets of tests are selected to build models for making pass/fail decisions. Two approaches are considered. A preemptive approach selects correlated tests to construct multivariate test models to screen out outliers. This approach does not rely on known customer returns. In contrast, a reactive approach selects tests relevant to a given customer return and builds an outlier model specific to the return. This model is applied to capture future parts similar to the return. The study is based on test data collected over roughly 16 months of production for a high-quality SoC sold to the automotive market. The data consists of 62 customer returns belonging to 52 lots. The study shows that each approach can capture returns not captured by the other. With both approaches, the study shows that multivariate test analysis can have a significant impact on reducing customer return rates especially during the later period of the production.

1 Introduction

A customer return is a part that passes a comprehensive test flow but fails on the customer's side. The root cause of these customer returns can be due to various issues such as insufficient testing (i.e. test escape), latent defect mechanisms, packaging issues, etc. The ultimate goal for high-quality products, such as those sold in the automotive market, is to have zero customer returns. In this market, a customer return denotes a rare event as the defective parts per million (DPPM) for the product line is extremely low. When a customer return occurs, it is analyzed carefully by the device or product engineer and modifications are made to improve the test and/or manufacturing flow. Since each return passed a comprehensive set of tests, it can be challenging to reproduce their failure mechanism with any individual test. Often, this analysis is aided by failure analysis (FA) to uncover the root cause of the customer return.

Each part passes all tests individually in a comprehensive set of tests before being shipped. For additional screening, a logical choice is to employ multivariate test analysis, where multiple parametric tests are used to build test models and dies outlying in these mutlivariate spaces are screened by these models, e.g. outlier models. To apply multivariate test analysis, one challenge is to select the relevant combinations of tests to use [8]. This is necessary when there are many parametric tests, potentially over 1000, and the possible test combinations can be enormous.

In this work, we study two multivariate test analysis approaches differentiated by how they select the tests. Figure 1 illustrates these two approaches.

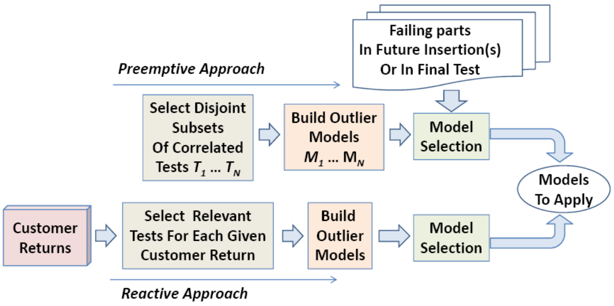


Figure 1. **Two multivariate test analysis approaches**

In the preemptive approach, tests are selected before any customer return is known. The work in [1] shows that outlier analysis performed with correlated tests can screen out defective parts. We adopt the same idea and select subsets of correlated tests $T_1, \dots, T_N$ such that tests within each subset T_i have a high mutual correlation, i.e. ≥ 0.9. Each subset T_i (with two or more tests) defines a *multivariate test space*, where outlier analysis can be performed.

With the N subsets, N outlier models $M_1, \dots, M_N$ can be built using various known techniques such as Principal Component Analysis (PCA) or one-class Support Vector Machines (SVM) [2]. Since each outlier model may incur some overkill, it may not be acceptable to apply all N models. Hence, a *model selection* step is required to select the outlier models that will be applied.

In model selection, if an outlier model is built based on test measurements in one insertion, the effectiveness of the model is evaluated using parts that fail subsequent wafer insertion(s) and final test (future fails). The effectiveness is reflected in two numbers: the number of known future fails

This work is supported in part by National Science Foundation, Grant No. 0915259 and Semiconductor Research Corporation, project 2010-TJ-2093.

978-1-4673-1595-1/12/$31.00 ©2012 IEEE

captured by the model and the number of overkill. The objective is to select the models that identify the most known future fails while minimizing the amount of overkill. In this study, we show that a preemptive approach can capture up to 15 returns. We also show practical results when the number of overkill is limited to $\leq 1\%$ of all parts.

In the reactive approach, a known return is given. A test selection step [8] is performed for the return. Then, an outlier model is built specific to the return. The objective of such a model is to capture parts whose multivariate signatures are similar to the known return. The model also goes through a model selection process to determine if it is effective to apply. If it is, the model is added to the pool of models to be applied in production. With the reactive approach, we show that 7 models learned from 7 known returns can capture 7 other future returns.

This study is based on test data collected over roughly 16 months of high-volume production for a SoC product in the automotive market. The DPPM for this product is close to zero. A large portion of the design is flash and smaller analog blocks. The data consists of parametric test measurements from three wafer sort insertions and final test, which target the flash and analog blocks. These tests include flash, current, voltage, conductivity measurements, etc. We have 62 customer returns belonging to 52 lots, where each lot contains one return and more than 12K passing parts. Associated with each lot is its production date. We also have the test date and return date for each return. To facilitate the discussion, the remainder of the paper will use labels R1 . . . R62 to denote the returns based on the date they were tested in chronological order.

The rest of the paper is organized as the following. Section 2 reviews prior works of multivariate parametric test analysis for improving quality. Section 3 discusses the preemptive approach and the results. Section 4 discusses the reactive approach and explains its differences from the preemptive approach in terms of the tests selected in multivariate analysis. Section 5 discusses the results when the preemptive and reactive models are applied to the 52 lots of data. Section 6 concludes.

2 Prior related works

The analysis of I_{DDQ} tests were shown to be able to identify defective parts including those susceptible to fail during burn-in and potential customer returns [15, 16]. As transistor geometries scale down, the leakage current increases which renders I_{DDQ} tests less effective [12]. There have been many extended I_{DDQ} tests [13, 14, 16], but these enhancements are not resistant to the effects of continuous scaling. This motivates the need for a more advanced statistical analysis approach that analyzes a wide range of tests such as flash and analog test measurements.

Many works have applied multivariate analysis on parametric test data and it has been suggested that it is an effective approach to improve the quality of traditional parametric testing [4]. This section reviews a few examples to show that various ideas have been proposed in the field.

The work in [1] applied Principal Component Analysis (PCA) to explore test correlations by mapping a high-dimensional test space into a low-dimensional space consisting of Principal Components (PCs). In the PC space, constructed using I_{dd} and analog measurements, the failing parts are shown to be outliers. Test limits can be applied in the PC space to screen out those failing parts. PCA can also be applied for dimension reduction as shown in [5]. The authors proposed a screening strategy based on the analysis of top PCs and demonstrated the ability to identify parts that were likely to fail burn-in.

Binary decision forests were applied in [6, 7] to predict devices that were likely to fail among a group of test insertions. In these works, parametric test measurements from 3 final test insertions were analyzed and the authors identified redundancies and suggested a method to replace the expensive tests with less expensive ones.

The work in [9, 11] analyzed a set of parametric wafer probe test measurements in order to understand what it takes to screen out customer returns using outlier analysis. It was shown that multivariate outlier analysis was more robust and effective at screening customer returns than traditional test limits. In [8], it was found that test selection is a critical step in the learning for screening customer returns. A methodology was suggested that leveraged various test selection algorithms to improve the ability to screen returns.

A forward prediction methodology was derived in [10] that studied the potential of using parametric wafer probe test measurements to predict parts that fail at final test and on the customer's side. When predicting customer returns, the authors used PCA on a set of important tests that best describe the customer return's failing behavior and an outlier model was built to screen out returns.

This work employs two learning techniques previously suggested: (1) applying PCA to correlated tests proposed in [1] and (2) applying PCA in conjunction with SVM [2][3] one-class outlier analysis proposed in [10]. The goal is to develop a methodology to apply these learning techniques before and after examples of customer returns become available. The focus is not optimizing a particular learning technique. Instead, the focus is on understanding the important considerations in developing a methodology and more interestingly, on understanding how effective a multivariate analysis approach can be for capturing customer returns.

3 The Preemptive Approach

In multivariate test analysis, the most important step is to decide on the subset of tests to use. For example, in our data there are more than 1000 parametric wafer probe tests. In the preemptive approach, we divide these 1000+ tests into disjoint groups based on correlation, where each group has a mutual correlation ≥ 0.9. These pairwise correlations are calculated based on the test measurements using only parts that passed all three wafer sort insertions from the earliest of the 52 lots. In total, there are 160 groups with 2-5 correlated tests and 370 tests are included in these 160 disjoint groups.

When considering the lower limit on the correlation, we saw no hard cut-off point in the number of test sets compared to the amount correlation. Hence choosing .85 correlation will result in similar groups of correlated tests.

3.1 Build an outlier model in a PC space

Each test in a group belongs to the same test insertion. Together, the group of tests defines a multivariate test space. A lot-based outlier model is built in this test space with the objective of identifying as many defective parts as possible while minimizing the amount of overkill.

As an example, a test space consisting of two highly correlated flash tests is shown in Figure 2. In this test space, there are two types of outlying behavior. A part can be outlying in the direction of the linear trend (PC1). These parts are seen in the top right and bottom left of the test space. Alternatively, a part can be outlying in the direction orthogonal to the linear trend (PC2). These parts are seen in the top left and bottom right of the test space.

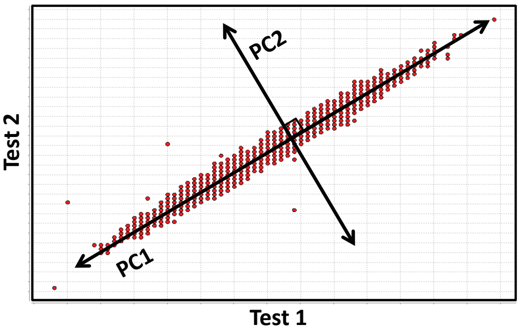


Figure 2. Two dimensional test space demonstrating the outlying directions along with the two PCs

Both outlying behaviors can be identified using PCA. PCA transforms the original test space into a PC space as shown in Figure 3. The first PC describes the direction of the linear trend, i.e. the direction having the most variance. The second PC is pointed in the direction orthogonal to the first PC.

In this PC space, an outlier model can be learned but for simplicity, test limits are set in each PC. In the study we use $\pm 3\sigma$ as the test limit in the first PC and $\pm 6\sigma$ for the remaining PC(s). Any part that falls beyond this range in the PC space is considered an outlier. Essentially, these limits define a box bounding the passing region in the PC space as shown in Figure 3.

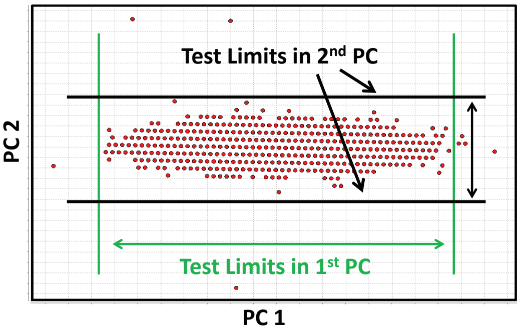


Figure 3. Applying test limits in the PC Space

In the direction of the first PC we apply a $\pm 3\sigma$ test limit to identify the marginal parts in the direction of the most variance. In directions with less variance, we apply $\pm 6\sigma$ limits to identify the gross outliers.

When applying the test limits to each group of tests, the parts that failed in the original test flow were removed so the resulting dataset did not have any gross outliers (to avoids biasing of mean and sigma). For each test, the mean of the data is centered at zero and the data is normalized by the standard deviation. The normalized data is transformed using PCA and the $\pm 3\sigma$ and $\pm 6\sigma$ limits are applied. Parts outside the tests limits are considered outliers. This was repeated for each of the 160 groups of correlated tests.

3.2 Number of overkill by a model

The average number of overkill per lot, for each of the 160 models, is sorted and shown in Figure 4. We define an overkill as a part that passes all tests in the original test flow and is not a customer return, but is screened by the model. We see that many models have fewer than 20 overkill per lot, which is less than 0.17% overkill rate per lot. Individually, this may seem insignificant but collectively, the overkill rate can add up quickly as more models are applied.

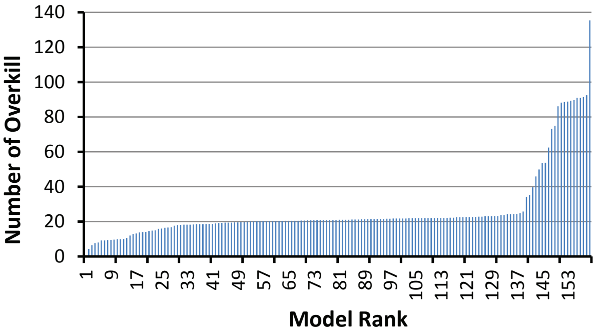


Figure 4. Avgerage # of overkill per lot by each model

Table 1 shows the first 10 models in Figure 4. The second column of the table shows the average overkill per lot. Out of these 10 models, only one model captures a customer return. If we add the average overkill rates per lot without considering the overlap, this gives 73.8 parts per lot or ~0.6%.

Model Num.	Avg. overkill per lot	Return Captured
M1	0.5	0
M2	4.3	0
M3	6.5	R4
M4	7.6	0
M5	8.0	0
M6	9.2	0
M7	9.2	0
M8	9.4	0
M9	9.5	0
M10	9.6	0

Table 1. **Top 10 models selected based on overkill rate**

For extremely high quality products, capturing a single customer return is valuable as it will help push the DPPM rate to zero. The result in Table 1 shows that one of the earliest return R4 is captured. To capture more customer returns, we need a method for selecting models that is more effective at identifying models that can screen returns.

To facilitate the discussion, we name the 160 preemptive models as M1 . . . M160, following the rank shown in Figure 4. Hence, model M1 has the smallest number of overkill and M160 has the largest.

3.3 Model selection based on future fails

In addition to the overkill, it is interesting to evaluate the effectiveness of the model with respect to capturing parts that fail in future wafer insertion(s) and package test. This ”forward prediction” capability can be an indicator of the ability to predict a customer return, which can also be considered a future fail. Figure 5 illustrates this forward prediction evaluation.

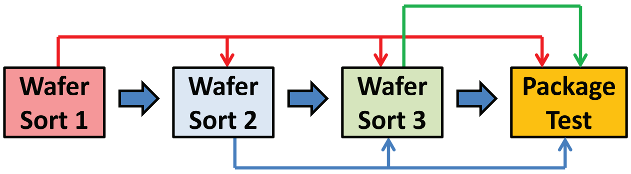


Figure 5. **Learning a model in one test insertion and using failing parts in future insertions to evaluate the effectiveness of the learned model**

As shown in Figure 5, the test flow consists of three wafer insertions and a package final test. To illustrate the concept of forward prediction, we take a group of tests from wafer sort 1. This model is evaluated using failing parts from wafer sort 2, sort 3, and package test. Similarly, a model built with tests in wafer sort 2 is evaluated using failing parts from sort 3 and package test. Note that models are built with only parametric wafer probe tests only (no package tests are included).

Figure 6 shows the average number of future fails per lot captured by each of the 160 models. In this figure, the models are sorted based on the number of future failures they screens out. These model numbers are different from those shown in Figure 4. Observe that many models can capture many future fails. For example, over 116 models that can screen 7 or more future fails.

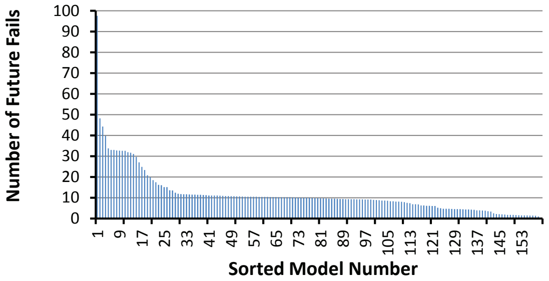


Figure 6. **Average number of future fails per lot screened by each model**

Applying all 116 models will result in an overkill rate too large to afford. If we assess the effectiveness of a model based on only the number of future failures screened can be misleading. Models that have more overkill inherently screen more future fails. Therefore, a simple heuristic is to consider the ratio of screened future fails over the incurred overkill. Figure 7 shows this ratio for the 160 models. Note that the models are sorted based on the ratio and the model numbers are different from the previous figures.

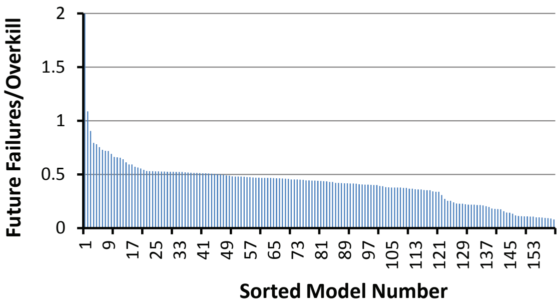


Figure 7. **The ratio of the number of screened future failures over the number of overkill by each model**

This ratio provides a more balanced approach to measuring the effectiveness of the outlier models. For example, setting a lower limit on the ratio to 0.5 results in 49 models.

3.4 The model selection heuristic

The outlier models are filtered with different thresholds. First, only the models that screen out at least 7 future failures are considered. Second, only the models that result in less than 50 overkill are considered. Third, only models with a ratio $\geq$ 0.5 in Figure 7 are included. This leaves us with 40 models.

We sort the 40 models based on the ratio of future failures over overkill and select the top n models. Table 2 shows the top 10 models with the number of overkill and the number of future fails captured by the models. In this table, the last column shows the accumulated overkill (total overkill when applying the top n models).

As mentioned before, the model names are those defined based on their ranks in Figure 4. The ”Test Count” is the number of tests used in each model. The ”Overkill” is the average number of overkill per lot. The ”Future Fails” is the average number of future fails captured per lot.

Model Name	Test Count	Over-kill	Future Fails	Returns Screened	Return Num.	Accum. Overkill
M3	4	6.5	13.5	1	R4	6.5
M4	3	7.6	8.3	0		14.1
M68	2	20.4	18.5	1	R43	34.5
M60	2	20.3	16.1	0		54.7
M49	2	19.4	15.1	0		74.1
M43	2	18.6	13.6	0		92.7
M17	2	12.9	7.9	1	R59	105.6
M38	3	18.3	11.1	2	R1, R7	123.9
M33	2	18.0	10.7	0		141.9
M44	3	18.7	11.1	1	R1	160.6

Table 2. The top 10 models after filtering out the less effective models and sorting models based on the ratio of future failures vs. overkill

In Table 2, we see that five returns can be captured with the top 8 models. The total accumulated overkill rate is 123.9 parts per lot or $\sim 1\%$.

3.5 Models that can capture customer returns

In total, there are 28 models that can capture at least one customer return. Among these 28 models, 12 models capture the return R1, the earliest return. This shows that the earliest return is also the easiest to capture. For the remaining 16 models, Table 3 summarizes their statistics.

Without considering overlap, the overkill rate for all models totals 283.1 parts per lot, or on average 17.69 parts per model per lot. The total overkill rate is about 2.36%.

First, observe that there are models that cover the same returns. For example M38 and M46 (captures R1 and R7) and as another example M32 and M68 (captures R43). Model M73 also captures R7. Hence, we only need 13 models to cover all 15 returns.

Table 3 illustrates the optimization objective for model selection, i.e. selecting models to cover all 15 returns with minimal overkill. In the preemptive approach, a model selection method is not aware of any known customer return. Hence, it needs to rely on other information to select the models such as the statistics used in the simple selection heuristic discussed above. An alternative approach is to have each model examined manually by an expert. For example, each model can be visualized in 2-3 dimensional PC space and an expert can decide if the group of tests make sense and/or if the outliers should be screened out or not.

Model Name	Test Count	Returns Screened	Overkill	Future Fails	Return Name
M3	4	1	6.5	13.5	R4
M17	2	1	12.9	7.9	R59
M21	2	1	14.0	2.0	R54
M27	2	1	15.9	1.6	R18
M28	2	1	16.0	8.2	R15
M32	2	1	17.6	1.8	R43
M38	3	2	18.3	11.1	R1, R7
M40	2	1	18.5	8.6	R44
M42	2	1	18.6	7.5	R32
M46	2	2	19.1	10.0	R1, R7
M47	2	1	19.3	9.3	R55
M64	2	1	20.3	8.9	R46
M68	2	1	20.4	18.5	R43
M73	3	1	20.6	9.9	R7
M114	3	2	22.0	3.8	R3, R10
M133	2	1	23.1	10.8	R12

Table 3. 16 models can capture 15 returns

Table 3 can be used as a way to select models. In this case, it no longer is a preemptive approach. Instead, a model is selected only if it captures a known return. For example, assume we want to target the last 20 returns. In the table we see that six of the last 20 returns can be captured by six models, i.e. models M17, M21, M32, M40, M47, and M64 capture returns R59, R54, R43, R44, R55 and R46 respectively. Assuming we cannot afford to apply all 13 models, then the six models may be preferred because they target later returns. In this case, the accumulated overkill for these six models is 102.6 parts per lot or $\sim$0.85%. This may be a more acceptable strategy to implement in practice.

3.6 Visualizing model M3

The model M3 is built using 4 highly correlated current tests transformed into the PCA space and the top 2 PCs are shown in Figure 8. In the PC space, $\pm 3\sigma$ limits are shown in the first PC and $\pm 6\sigma$ limits are shown in the remaining three PCs.

In Figure 8, A rectangle is created by the test limits (red) that encloses a passing space and everything outside the rectange is considered an outleir. As we can see, most parts residing in the outlier space are failing. This model captures 1 customer return R4 which is marked by the relatively large green dot. The model also captures on average 58.4 parts per lot that fail in the same test insertion, 13.5 future failures per lot and 6.4 overkill per lot. We see that those overkill fall into the clusters of failing parts that either fail in the insertion or in the future. One can argue that those 6.4 overkill are marginal parts and should be screened out any way. Overall we see that the outlier model is quite effective. Manual analzying the model increases our confidence and

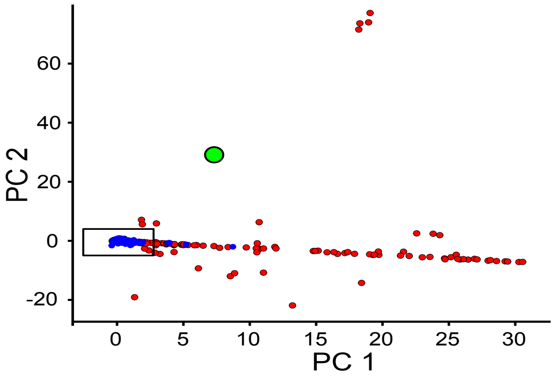


Figure 8. Two dimensional PCA space where a customer return is seen as an outlier

justifies applying the model in production.

4 The Reactive Approach

In the reactive approach, a known customer return is used to learn a model. Figure 9 describes the learning flow.

Given a customer return, a test ranking is obtained for all parametric wafer probe tests. The ranking is based on the *outlier rank* of the return in each test. An outlier rank for a test is defined as the distance to the mean of the distribution, where the mean is computed using only passing parts in one lot. For example, an outlier rank 20 means there are 19 passing parts further away from the mean than the given return. All other parts are closer to the mean than the return.

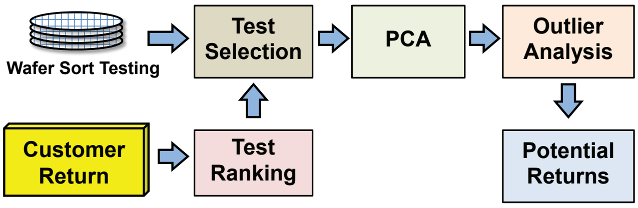


Figure 9. The reactive learning flow

For a return, each test is associated with an outlier rank and the tests are ranked using the outlier rank, where a test is considered more important if the outlier rank is lower (implying more outlying behavior). Based on the test ranking, the top 10 tests are selected. The parametric data of all parts that passed wafer sort testing are transformed using PCA. In the PCA space, an outlier model is built using the one-class ν-SVM algorithm[10]. In this algorithm, the value ν represents the upper bound on the fraction of outliers identified by the model (The actual number of outliers can be much smaller) [2]. In the study, we set $\nu = 1\%$, meaning that each model would have no more than 1% parts as outliers.

In the rest of the section, the discussion centers on three main ideas: (1) How the reactive approach screen customer returns. (2) Why the reactive approach captures a different set of customer returns than the preemptive approach (3) Why it is important to apply PCA before building a one-class SVM outlier model.

4.1 Returns captured by the approach

The discussion will focus on models built from customer returns R1 . . . R42. We name the resulting models MR1 . . . MR42, respectively. Models can be built for returns R43 to R62, but these parts were returned after all the parts in our dataset had shipped. Hence, we could not validate these models using the returns in our dataset. As a result, we will not build a model nor discuss these returns.

Table 4 shows the top 18 models ranked by the column ”Corr. Tests”. This ranking and the ”Corr. Clusters” will be explained shortly (Section 4.3). The column ”Model Name” shows the model name corresponding to the return name. The ”Overkill” is the number of overkill for capturing the return(s). The number of overkill does not change significantly across lots and can be used as an estimate of the amount of overkill per lot.

Model Name	Over-Kill	Corr. Tests	Corr. Clusters	Return Captured
MR8	12	9	3	R50
MR18	11	8	2	R62
MR11	8	8	1	R45
MR1	7	7	2	
MR12	8	7	2	
M7	8	7	3	
MR16	1	6	3	
MR13	6	6	3	R57
MR38	12	6	2	
MR37	37	6	3	R53
MR25	15	6	3	
MR5	34	6	2	R58
MR21	0	5	2	
MR26	0	5	2	
MR10	4	5	1	
MR41	5	5	2	
MR24	43	4	2	R43
MR3	3	4	2	

Table 4. Top 18 models following a ranking based on the correlated tests selected to build each model

Table 4 shows that the top 18 models can capture 7 future returns. Interestingly, all 7 returns fall into the category of the last 20 returns. The total number of overkill shown in this table is 214 parts per lot, about 1.8% of the total die population. Suppose we are limited to 0.5% overkill (∼60 parts per lot) for the reactive approach. We could select the top 8 models, which would capture four future returns with 61 overkill per lot.

4.2 The timeline view of the result

Figure 10 shows a timeline of the 7 models in Table 4 that captures a future customer return. In the figure, the green triangle pointed downward represents the shipped date of a part that would eventually fail in the field. There is a period of time before the part is actually returned. This period of time is represented by the light blue bar. When the part fails and is returned, the date is shown as the yellow triangle pointed upward. At this time, it is possible to learn an outlier model for the customer return and apply the resulting model to future lots. As a result, parts that would become customer returns in the future could be screened out. This is shown as the red arrow pointed downward which marked the test date of this future customer return.

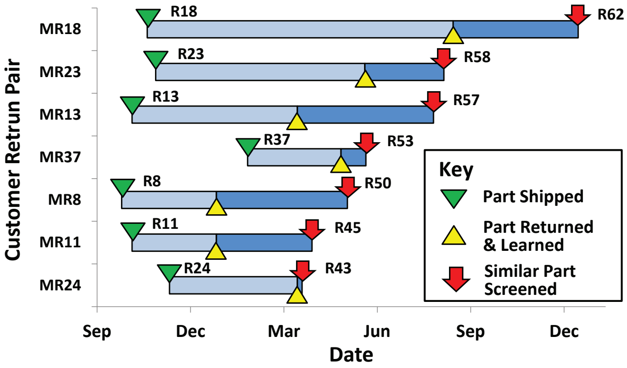


Figure 10. The timeline view

The minimal turn-around time required for the learning and applying an outlier model is 5 days, which is determined by the time between the return date of R24 and the test date of R43.The next shortest turn-around time is more than 3 weeks between the date R37 is returned and the test date of R53. Hence, if learning and applying an outlier model in production requires more than a week, we would only miss R43 while catching the other six returns. This timeline gives an idea on the required turn-around time to implement the reactive approach.

4.3 Challenges of the test selection when test correlation is ignored

To understand why the number of correlated tests is used to rank models in Table 4, we need to first explain why a model learned from a return has the ability to capture another return and why the captured return is not captured by the preemptive approach earlier.

We begin by explaining that the ability to capture a future return largely depends on selecting the correct combination of tests to build the model. We will show that selecting the correct set of tests is challenging when ignoring test correlation and we also show that building models in the test space is not effective.

Take the first model, MR8, in Table 4 as an example. Figure 11 shows how the return R8 (red) behaves for the top 10 tests (A to J) based on its outlier rank. Also shown is the behavior of the return R50 (green) that is captured by the model learned from R8.

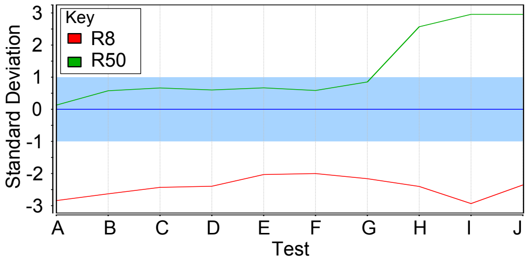


Figure 11. Deviation of Customer Return Pair (R8 and R50) in the top 10 tests based on the outlier ranks of R8

In this figure, the parts that failed wafer level testing were removed and then, each parametric test is normalized to zero mean and unit standard deviation. The normalization values for each test (mean shift and standard deviation) based on the lot containing R8 are used to normalize the lot containing R50. In Figure 11, the normalized test measurements for both returns are shown on the y-axis in terms of the standard deviation. Also shown is the $\pm 1\sigma$ band (blue).

As we can see, return R8 deviates from the $\pm 1\sigma$ band in all 10 tests. In contrast, R50 resides in the $\pm 1\sigma$ band for tests A-G and only deviates from the band in tests H, I, and J. Hence, R8 shows outlying behavior in all 10 tests while R50 shows outlying behavior in three tests, H, I and J.

Num. of Tests Used	Rank Based on R8			Re-Rank Tests Based on R50		
	Test	Overkill		Test	Overkill	
		R8	R50		R8	R50
1	A	19	5814	H	34	2
2	D	7	5242	J	55	35
3	H	3	76	I	67	4
4	B	5	128	G	25	5
5	E	4	227	F	22	5
6	G	3	293	B	12	18
7	I	4	130	E	9	29
8	F	6	190	C	13	44
9	C	7	226	D	11	71
10	J	7	87	A	7	87

Table 5. The test selection problem

In Table 5, we show the results when we apply SVM one-class to build an outlier models using various sized test sets (without using PCA). Columns 2-4 show the results when building outlier models using the outlier rank for R8 (column 2). Tests A-J are sorted based on the outlier rank of R8. For example, test A is ranked 1, test D is ranked 2, test H is ranked 3, etc.

Two overkill columns are shown. The ”R8” overkill column shows the number of overkill incurred by a model that screens out R8. This model is built using the top n tests based on the outlier test ranking for R8. For example, a model learned from the top 4 tests (A, D, H, and B) can screen out R8 while incurring 5 overkill. Using the same set of tests, an outlier model is learned that screens out R50 in its respective lot that results in 128 overkill (”R50” overkill Column).

If we use the test ranking given by R8 and if the upper limit on the number of overkill per lot is 60 (i.e. $\leq 0.5\%$ overkill), then none of the 10 models in Table 5 would be able to capture return R50. However, this does not imply that a model built using the same subset of tests (not following the outlier ranking of R8) cannot capture R50.

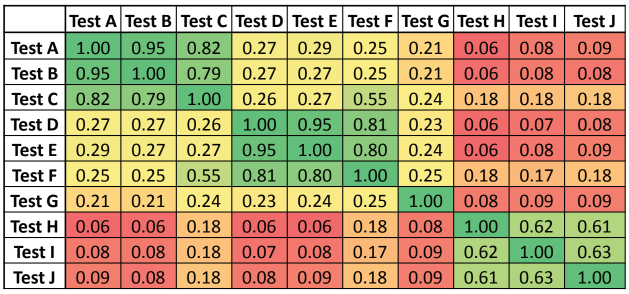
The same subset of tests are re-ranked based on the outlier ranks of R50 and the same experiment is performed with the new test ranking. The results of this experiment are shown in the last 3 columns of Table 5. We see that if we select the top 6 tests (H, J, I, G, F and B), then we can build a model to screen R8 with 12 overkill and R50 can be captured with 18 overkill in its respective lot. In other words, the outlier model built from R8 can capture R50 if we select the correct combination of tests.

This illustrate the difficulty of test selection. In the following section, we will explain how PCA can alleviate this difficulty and it makes it possible for a more effective model to be built.

4.4 Correlations in the top 10 tests

Figure 12 shows the correlation matrix for test A-J. The correlations are calculated based on measured values of all passing parts in the lot containing R8. The matrix is colored where the highly correlated clusters of tests are shown in green and the uncorrelated tests are shown in red.

Observe that among the 10 tests, there are three *correlated clusters*. This number is shown under the column ”Corr. Clusters” in Table 4. Due to the existence of correlation and clusters, PCA can be applied to explore the correlation structure.



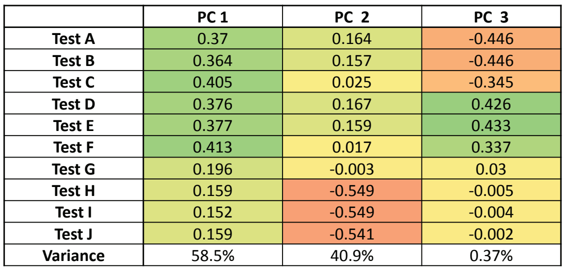
	Test A	Test B	Test C	Test D	Test E	Test F	Test G	Test H	Test I	Test J
Test A	1.00	0.95	0.82	0.27	0.29	0.25	0.21	0.06	0.08	0.09
Test B	0.95	1.00	0.79	0.27	0.27	0.25	0.21	0.06	0.08	0.08
Test C	0.82	0.79	1.00	0.26	0.27	0.55	0.24	0.18	0.18	0.18
Test D	0.27	0.27	0.26	1.00	0.95	0.81	0.23	0.06	0.07	0.08
Test E	0.29	0.27	0.27	0.95	1.00	0.80	0.24	0.06	0.08	0.09
Test F	0.25	0.25	0.55	0.81	0.80	1.00	0.25	0.18	0.17	0.18
Test G	0.21	0.21	0.24	0.23	0.24	0.25	1.00	0.08	0.09	0.09
Test H	0.06	0.06	0.18	0.06	0.06	0.18	0.08	1.00	0.62	0.61
Test I	0.08	0.08	0.18	0.07	0.08	0.17	0.09	0.62	1.00	0.63
Test J	0.09	0.08	0.18	0.08	0.09	0.18	0.09	0.61	0.63	1.00

Figure 12. **Correlation Matrix of the top 10 tests**

Also notice that if we apply the 0.9 (90%) correlation threshold as we did in the preemptive approach to extract groups of tests, then tests A and B will be in one group and test D and E will be in another group. All other tests will be discarded. This shows that the reactive approach selects tests that are quite different from those considered in the preemptive approach.

4.5 Apply PCA to explore correlated tests

The dataset consisting of the tests A-J was transformed using PCA and the eigenvectors for the top 3 PC are shown in Figure 13. These eigenvectors describe the direction of each PC. For example, the first PC is mostly pointed in the direction of tests A-F as the eigenvalues for these tests are larger. Notice that tests A-F belong to the first two correlated clusters shown in Figure 12. The second PC is mostly pointed in the direction of tests H-J which belong to the third correlated clusters shown in Figure 13. This figure also shows the % of variance accounted by each of the top 3 PCs. Notice that the first two PCs account for 99.4% of the total variance.



	PC 1	PC 2	PC 3
Test A	0.37	0.164	-0.446
Test B	0.364	0.157	-0.446
Test C	0.405	0.025	-0.345
Test D	0.376	0.167	0.426
Test E	0.377	0.159	0.433
Test F	0.413	0.017	0.337
Test G	0.196	-0.003	0.03
Test H	0.159	-0.549	-0.005
Test I	0.152	-0.549	-0.004
Test J	0.159	-0.541	-0.002
Variance	58.5%	40.9%	0.37%

Figure 13. **Eigenvectors and variance accounted**

In Figure 14, we project the measured values of R8 and R50 onto the three PCs. The three PCs were normalized to zero mean and unit standard deviation. As we can see, both R8 and R50 deviate from the mean in the first two PCs. More specifically, R8 deviates more in the first PC and R50 deviates more in the second PC.

Recall that in Figure 11, R50 deviates only on tests H, I, and J. In Figure 12, we see that H, I, and J belong to the third correlated cluster. In Figure 13, we see that the second PC mostly pointed in the direction of tests H, I, and J. Hence, it is not surprising to see that R50 deviates more in PC 2 in Figure 14.

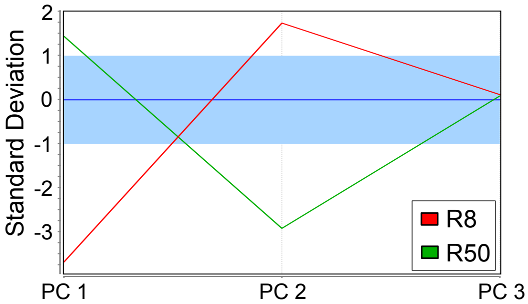


Figure 14. **Behavior of R8 and R50 in the top 3 PCs**

SVM outlier models are built using sets of the top 3 PCs. The overkill required to screen out each return is shown in Table 6. We see that using the first two PCs, an outlier model can screen out R8 with 4 overkill and R50 with 12 overkill. We see that PCA alleviates the difficulty of test selection, as discussed above, by transforming set of 10 correlated tests into fewer PCs. It is much easier to select the top PCs because the higher PCs (e.g. PC 3, PC 4, ...) do not account for much of the total variance and can be discarded.

PCs in use	R8 Overkill	R50 Overkill
1	2	931
1,2	4	12
1,2,3	7	22

Table 6. Screening Returns R8 and R50 in PCA space

When building an outlier model in the PC space, the results in Table 6 suggests that we may only want to use the first PC as it results in the fewest overkill. Building an outlier model in 2 PCs is better because it accounts for 99.4% of the total variance and describes more of the test space.

The 2-dimensional space consisting of the top 2 PC is shown in Figure 15, where all passing parts are blue, R8 is marked as a green dot and R50 is marked as a red dot. As we can see, R8 and R50 are outlying in opposite sides of the distribution but both customer returns can be screened with outlier analysis.

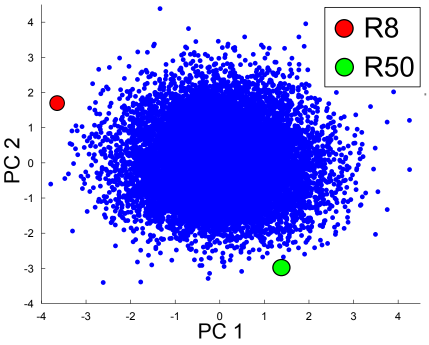


Figure 15. R8 and R50 seen in the top 2 PCs

4.6 Why we use the number of correlated tests to rank models in Table 4

Figure 12 shows three correlated clusters with 9 correlated tests. The number of correlated tests is also shown in Table 4, which we use to rank models. There are two thoughts driving us to use the heuristic.

Recall that the top 10 tests are selected for a return based on their outlier ranks, which means that the return is outlying the most in these tests. Suppose that among these 10 tests, i tests are correlated (they can form different clusters). When there are many correlated tests among the top 10 tests, fewer PCs are required to describe the test space. Because the top PCs can account for most variance, it is easier to identify and select them. For example, based on the variance in Figure 13, it is clear that the top two PCs should be selected and the third can be discarded. This simplifies the test selection.

In contrast, suppose the top 10 tests are not correlated. Then, PCA will produce 10 PCs each accounting for a significant portion of the variance. In this situation, we run into the difficulty of the test selection as discussed in Table 5 before. Then, PCA essentially becomes ineffective at simplifying the test selection problem. Suppose we have 10 uncorrelated tests $t_1, \ldots, t_{10}$. Suppose a future return can only be captured based on the model built on t_2, t_4, t_7 but the return used in the learning can be screened out with any combinations of three to five tests. Before seeing the future return, the test selection somehow needs to figure out it is the combination t_2, t_4, t_7 that is desired. This would be a difficult problem to solve.

The results shown in Table 4 confirm the effectiveness of the model ranking heuristic. There are 42 models built for R1 to R42. Only seven of them can capture a future return with a reasonable number of overkill, i.e. ≤ 60 parts. We see in the table that the top 12 models contain the six that can screen future returns.

5 Applying Both Preemptive and Reactive Approaches

The preemptive and reactive methodologies were applied to the 52 lots and the returns screened are shown in Figure 16. In this figure, the 62 returns are ordered by their test dates which spanned a 16 month period. Based on the test date, returns R1-R42 were tested in the first 6 months and returns R43-R62 were last tested in the remaining 9 months. We separate the first 42 from the last 20 returns and we will focus on predicting the last 20 returns.

Assuming that we apply all 16 preemptive models in Table 3 and 7 effective reactive models in Table 4, it is possible to screen out 21 returns. These returns are shown in Figure 16, where returns captured by the preemptive approach are green triangles, returns captured by the reactive approach are red squares, and all returns not screened in this study are blue diamonds. The preemptive approach can screen 15 returns where only one return (R43) is among the 7 that are captured by the reactive approach. Hence, only 14 markers are shown for the preemptive approach in Figure 16. This also shows that the two approaches complement each other.

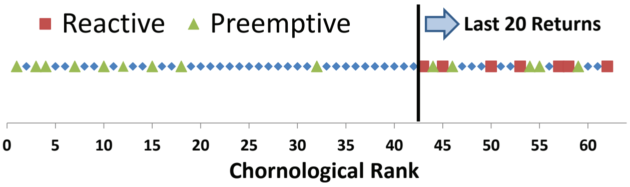


Figure 16. Summary of chronological results

Besides screening as many returns as possible, another objective is to target the last 20 returns. In Figure 16, we can screen 12 of these returns with both approaches. These returns are more challenging to capture. Every part that is returned is analyzed and when possible, new test(s) or process modifications are made to prevent similar types of returns from occurring. Despite these modifications, test escapes still occur. Hence, capturing a later return would be more significant. This also shows how preemptive and reactive methods complement the existing test methodology.

Figure 16 shows the ”optimal” scenario where the two model selection steps are assumed to be optimal in the sense that they would include the desired models to apply. With both approaches, there are models that do not capture any of the 62 returns in the study and those models will incur overkill. Hence, a model selection heuristic was employed with the objective of selecting models that capture future returns while minimizing the number of models that do not.

In a practical setting, the model selection is limited by the amount of overkill incurred by the collection of models. Assuming the accumulated overkill is limited to 1% for preemptive screening and 0.5% for the reactive screening, then we can select and apply the top 8 preemptive models in Table 2 and the top 8 reactive models in Table 4. The customer returns screened by these models are shown in Figure 17.

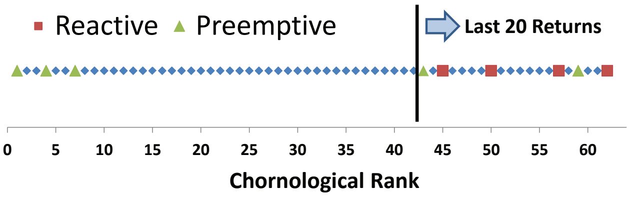


Figure 17. Results by simple model selection heuristics

In Figure 17, we see that 5 returns are captured with the preemptive method and 4 returns are captured with the reactive method. In total, 9 customer returns can be captured. Also observe that 6 of the last 20 returns can be captured. This shows that multivariate test analysis can have a significant impact to reduce the customer return rate in practice.

6 Conclusion

In this work, we present results based on two multivariate test analysis approaches, a preemptive approach and a reactive approach, for screening potential customer returns. The study is carried out using 52 lots of test data with 62 known customer returns. We show that the preemptive approach can capture up to 15 returns and the reactive approach can capture up to 7. In total, 21 customer returns can be screened where one return is captured by both approaches. If we target the last 20 returns, which are more challenging to screen with the existing testing methodology, we show that both approaches capture 12 of the last 20 returns. Our findings suggest that multivariate test analysis can make a significant impact to reduce the customer return rate in practice, especially during later periods of the production when customer returns are more sparse and harder to catch.

References

[1] Peter M. O’Neill. Production Multivariate Outlier Detection Using Principal Components. *IEEE International Test Conference*, 2008.

[2] B. Scholkopf and A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization,and Beyond. *The MIT Press*, 2001

[3] M. Tsai, C. Ho and C. Li, Active Learning Strategies Using SVMs. In *IJCNN*, 2010

[4] W. Robert Daasch, C. Glenn Shirley and Amit Nahar. Statistics in Semiconductor Test: Going beyond Yield. IEEE Design & Test of Computers, vol 26, issue 5, 2009, pp. 64-73.

[5] Amit Nahar, Robert Daasch and S. Subramaniam. Burn-in Reduction using Principle Component Analysis. *IEEE International Test Conference*, 2005.

[6] Biswas, S. and Blanton, R.D. Reducing Test Execution Cost of Integrated, Heterogeneous Systems Using Continuous Test Data. *Computer-Aided Design of Integrated Circuits and Systems*, vol 30, issue 1, 2011, pp. 145-158.

[7] Biswas, S. and Blanton, R.D. Statistical Test Compaction Using Binary Decision Trees *IEEE Design & Test of Computers*, vol 23, issue 6, 2006, pp. 452-462.

[8] N. Sumikawa, D. Drmanac, L. Winemberg, L. Wang and M. Abadir. Important Test Selection For Screening Potential Customer Returns. International Symposium on VLSI Design, Automation and Test, 2011.

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

[10] Nik Sumikawa, Dragoljub (Gagi) Drmanac, Li-C. Wang, LeRoy Winemberg and Magdy S. Abadir Forward Prediction Based on Wafer Sort Data. In *ITC*, 2011.

[11] Nik Sumikawa, Dragoljub (Gagi) Drmanac, Li-C. Wang, LeRoy Winemberg and Magdy S. Abadir Understanding Customer Returns From A Test Perspective. In *VTS*, 2011.

[12] J. Figueras and A. Ferre Possibilities and Limitations of IDDQ Testing in Submicron CMOS. *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, part B, Vol. 21, No.4, Nov. 1998, pp. 352-359.

[13] A. Miller IDDQ Testing in Deep Sub-micron Integrated Circuits. *ITC*, 1999, pp.724-729.

[14] P. Maxwell et al. Current Ratios: A Self-scaling Technique for Production IDDQ Testing. *ITC*, 1999, pp. 738-746.

[15] R. Kawahara, O. Nakayama, and T. Kurasawa The effectiveness of IDDQ and high voltage stress for burn-in Elimination. *IEEE Workshop on IDDQ Testing*, 1996, pp. 9-13.

[16] S.S Sabade and D.M. Walker ; Evaluation of effectiveness of median of absolute deviations outlier rejection-based IDDQ testing for burn-in reduction. *VTS*, 2002, pp. 81-86.