Feature Engineering with Canonical Analysis for Effective Statistical Tests Screening Test Escapes

Fan Lin, Chun-Kai Hsu, Kwang-Ting Cheng
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

Abstract—It is known that statistical analysis of test data can help screen potential test escapes without additional physical measurements. Based on analysis of production test data, this paper focuses on feature engineering for statistical tests to screen test escapes. The features are engineered in two aspects: development of effective features and transformation of features into different spaces in which the inherent differences between the test escapes and the normal population can be compacted into a small number of features.

In feature development, we generate two sets of features to characterize a chip based on the amounts of the chip’s test measurements deviated from the measurement means and their amounts deviated from the spatial patterns among dies on the same wafer. In feature transformation, the features are projected into the canonical space, in which the separation between the test escapes and the good chips are encapsulated into the first few dimensions. We show that each set of features reveals a unique set of test escapes, and the transformation of features can result in significant runtime reduction while keeping a comparable differentiating power as that in the original features. Therefore, both sets of features should be utilized and the canonical transformation should be applied when developing statistical tests for test escape reduction.

I. INTRODUCTION

For many applications, the requirements of the defective parts per million (DPPM) of integrated circuits have to be extremely close to zero. Each field return found at the customer side incurs significant cost and requires thorough analysis of the cause. It has been shown that a good fraction of field returns are test escapes that pass the complete test program, but fail at system level due to their intrinsic defects [1], [2]. Applying system tests to each chip prior to shipment, however, is undesired because it often results in high test time and cost.

This paper addresses the problem of identifying as many test escapes as possible by statistical analysis of the test data produced by a given test program, without taking any additional physical measurements. Such an approach can be viewed as adding statistical tests to the original test program [3]. Our main focus is on engineering novel features for statistical tests and demonstrating the importance of feature engineering for effectively capturing test escapes by statistical tests. Specifically, we develop features based on how a chip’s measurements deviate from the means of a set of normal chips and how a chip’s measurements deviate from the spatial patterns on a wafer. We then transform the features to a canonical space in which the separation between normal chips and test escapes of the projected data is maximized. The multivariate statistical approach based on these features incorporates both the inter-test-item correlations and the spatial correlations, and applying statistical tests based on the transformed features achieves significant runtime reduction based on standard classification algorithms. The proposed flow can be easily extended to include more sets of features and applied to a wide range of products.

To screen potential test escapes, one technique proposed by the Automotive Electronics Council is the part average testing (PAT) [4]. For some suggested electrical tests, PAT compares the measurement of a query chip with the mean of a set of normal chips and discards the query chip if its distance away from the mean is above a threshold. To address PAT’s limitation of evaluating individual test item only and ignoring the multivariate relation among test items, several other studies proposed multivariate screening approaches that incorporate the inter-test-item correlations.

O’Neill [5] applied outlier analysis with principal component analysis (PCA) on sets of correlated test items. Sumikawa et al. [1] extended O’Neill’s work with sophisticated model and test selection schemes and developed a preemptive and a reactive approach, depending on whether known field returns were given. Butler et al. [6] successfully demonstrated burn-in minimization by a collection of multivariate analysis. Chen et al. [2] showed various data mining techniques on final test data to predict system level test (SLT) failures.

In addition to inter-test-item correlations, there exist spatial correlations among dies on the same wafer. Stine et al. [7] modeled and decomposed spatial variations into four components: wafer-level variation, die-level variation, wafer-die interactions, and residuals. In capturing the spatial patterns with only a small amount of samples, Li et al. [8] proposed a virtual probe (VP) technique and Kupp et al. [9] proposed an estimation with a Gaussian process model. Nahar et al. [10] and Riordan et al. [11] used the spatial correlation of neighboring dies for defect prediction. Sumikawa et al. [12] identified abnormal wafers based on the spatial patterns of tests.

Taking into account the multiple correlations in test data, we also investigate a data transformation technique based on multivariate analysis of variance (MANOVA). MANOVA has been used in various fields to analyze the difference in the means of features between populations of samples [13], [14]. Based on MANOVA, a canonical analysis can be applied on the samples to form a set of canonical variables which can be linear combinations of the original test measurements. The linear combinations are chosen such that the first canonical
variable achieves the maximum separation between populations, the second canonical variable achieves the maximum separation between populations subject to it being orthogonal to the first canonical variable, and so on. Essentially MANOVA shows if there is a significant difference in the means between populations, and the canonical analysis could identify combinations of the variables to maximize the separation between populations. In this paper we utilize the canonical analysis to project the features into a canonical space for further statistical tests, which can more easily screen out test escapes with standard classification methods.

The rest of the paper is organized as the following: Section II illustrates how we develop features that represent different characteristics of a chip. Section III introduces the canonical analysis to further transform the features. The application of the proposed statistical tests is described in Section IV, and Section V shows experimental results on production test data. Section VI concludes the paper.

II. FEATURE DEVELOPMENT

In this paper, we use the residual vector of each chip as the base of input features for statistical analysis. In general, each chip with $N$ test measurements can be characterized by an $N \times 1$ residual vector $r$:

$$r = x_m - x_e$$

where $x_m$ is an $N \times 1$ vector of the measured values and $x_e$ is an $N \times 1$ vector of the estimated values.

The residual vector represents how the measurement values of a chip deviate from its estimated values. There are two aspects we can engineer such feature. Choosing different estimated values to calculate the residual vector may reveal different test escapes, and transforming the residual vector to another space may enhance the performance of the classifier.

We first explore the choices of estimated values for calculating the residual vector. The transformation of test data to another space will be discussed in Section III.

A. Measurement Mean

It has been shown that some test escapes differ from normal chips in their test measurement values relative to the measurement means [2]. Fig. 1 shows two wafer maps whose measurement values are standardized to the z-score by

$$z = \frac{x - \mu}{\sigma}$$

where $x$ is the measurement value, and $\mu$ and $\sigma$ are the mean and standard deviation of the measurements for chips on the same wafer, respectively. In Fig. 1a the circled chip is a test escape with a relatively abnormal feature value based on the difference between its measurement value and the measurement mean of the entire wafer. Note that Fig. 1a shows only one test item and thus only one out of $N$ features of the chips. There are other chips in the same dark blue region, which are as outlying as the circled chip is for this test item, but in the multivariate analysis they will not necessarily be classified as escapes.

(a) A test escape that is abnormal with respect to measurement mean

(b) A test escape that is abnormal with respect to spatial pattern

Fig. 1: Examples of test escapes which are considered abnormal with respect to different estimated values.

B. Spatial Pattern

Spatial patterns have been observed on wafers in many test items [7], [12], [15]–[17]. Fig. 1b shows a test escape which would be considered abnormal based on spatial pattern analysis. A wafer’s spatial patterns for some test items are the results of systematic variations which exist even without any additional manufacturing imperfections. Taking into account a test item’s spatial pattern and using the unique predicted value at each die location (derived from the test item’s learned spatial pattern) as the estimated value, we effectively eliminate the systematic variations and incorporate only the effects of other variations (including random variations) in the residual vector. In other words, with this revised residual vector which takes into account test items’ spatial patterns, we more accurately capture the noises in the wafer map images. In this paper, we employ bilateral filtering, a well-developed filtering technique in image processing, to denoise a wafer map and retrieve a systematic spatial pattern of a test item.

Bilateral filtering [18] is a non-linear filtering technique which extends the concept of Gaussian filtering to weight coefficients based on both relative spatial distance and pixel intensity difference. Pixels that are spatially close but have significant difference in intensity will have smaller weights, while pixels that are a bit farther apart but very similar in
intensity will have larger weights. Therefore, the sharp edges of an image can be preserved and the noises are more likely to be filtered. There are two kernels for evaluating the weights of the neighboring pixels. The domain kernel \( K_d \) evaluates the weights based on the spatial distance of pixels. The range kernel \( K_r \) evaluates the weights based on the pixel intensity difference.

Given the original image \( I \), pixel coordinates \( x \), and the filter window \( \Omega \), the filtered image \( I_f \) is defined as

\[
I_f(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) K_r(\|I(x_i) - I(x)\|) K_d(\|x_i - x\|)
\]  

where

\[
W_p = \sum_{x_i \in \Omega} K_r(\|I(x_i) - I(x)\|) K_d(\|x_i - x\|)
\]

For a \( P \times Q \) wafer map with values ranging from \( v_{\min} \) to \( v_{\max} \), we choose a Gaussian function with \( \sigma = \min(P, Q)/16 \) for \( K_d \), a Gaussian function with \( \sigma = (v_{\max} - v_{\min})/10 \) for \( K_r \), and the whole wafer map as the filter window \( \Omega \).

Fig. 2 shows the result of applying bilateral filter to one test item on a wafer map. Fig. 2a shows the original measurement of the test item. The residual of the measurement with respect to the wafer mean is shown in Fig. 2b, in which the spatial pattern is preserved. The residual of the measurement with respect to the bilateral filtered wafer map is shown in Fig. 2c, in which the spatial pattern in the original measurement is eliminated and the residuals better represent abnormality with respect to the spatial pattern.

Note that the circled test escape in Fig. 1a is abnormal considering its relatively large measurement value, but it is perfectly normal if we take into account the overall spatial pattern. The circled test escape in Fig. 1b is abnormal in the spatial pattern, but its measurement value is actually very close to the mean of the wafer. Therefore, each of the two test escapes shown in Fig. 1 can only be uniquely identified as abnormal, or potential test escape, by one of the two choices in selecting the estimated values for calculating residual vectors.

### III. Feature Transformation

Besides exploring two different choices of the estimated values for calculating the residual vectors to enrich the input features, projecting these features into different spaces before applying statistical tests may help improve the performance of classifiers. In this paper we introduce canonical analysis based on multivariate analysis of variance (MANOVA) to transform data into a canonical space in which the data of test escapes and normal chips are maximally separated in the first few dimensions.

Multivariate analysis of variance (MANOVA) [13] is a technique that, given \( g \) populations of samples in an \( N \)-dimensional space, compares the mean vectors of the populations and investigates which mean components differ significantly. Given the populations of samples:

\[
\begin{align*}
\text{Population 1 : } & x_{11}, x_{12}, \ldots x_{1n_1} \\
\text{Population 2 : } & x_{21}, x_{22}, \ldots x_{2n_2} \\
\vdots & \\
\text{Population } g : & x_{g1}, x_{g2}, \ldots x_{gn_g}
\end{align*}
\]  

where \( x_{ij} \) is an \( N \times 1 \) vector of the \( j \)th sample in the \( i \)th population, and \( n_i \) is the number of samples in the \( i \)th population.
Each observation $x_{ij}$ can be decomposed into three components: overall sample mean $\bar{x}$, the population effect $(\bar{x}_i - \bar{x})$, and the residual $(x_{ij} - \bar{x}_i)$, and

$$x_{ij} = \bar{x} + (\bar{x}_i - \bar{x}) + (x_{ij} - \bar{x}_i) \quad (6)$$

Subtracting $\bar{x}$ from both sides of (6) and summing the cross products over $i$ and $j$ yields

$$\sum_{i=1}^{g} n_i \sum_{j=1}^{g} (x_{ij} - \bar{x})(x_{ij} - \bar{x})' = \sum_{i=1}^{g} n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})' + \sum_{i=1}^{g} \sum_{j=1}^{g} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)' \quad (7)$$

or expressing it as:

$$S = B + W \quad (8)$$

where

$$S = \sum_{i=1}^{g} n_i \sum_{j=1}^{g} (x_{ij} - \bar{x})(x_{ij} - \bar{x})'$$

$$B = \sum_{i=1}^{g} n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})' \quad (9)$$

$$W = \sum_{i=1}^{g} \sum_{j=1}^{g} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)'$$

Eq. (8) shows that the total variance $S$ is the sum of the between-population variance $B$ and the within-population variance $W$. After such decomposition, MANOVA investigates if there exists significant difference between the population mean vectors using metrics based on $B$ and $W$. For example, if the Wilks’ lambda

$$\Lambda = \frac{|W|}{B + W} \quad (10)$$

is too small, we can conclude that there exists significant difference between the populations.

After MANOVA, the canonical analysis suggested in [14] could be used to create a set of canonical variables which are the linear combinations of the original variables. The criteria for choosing the linear combinations are that the first canonical variable should exhibit the maximum separation between the populations, the second canonical variable should be orthogonal to the first canonical variable while also exhibit maximum separation between the populations, and so on.

The process of generating the canonical variables in canonical analysis is similar to generating the principal components (PCs) in principal component analysis (PCA) [13]. PCA is a feature reduction technique that generates a set of new features, named principal components, which are mutually orthogonal and are ordered by the amount of variability in the data each PC explains. Given a set of data in matrix $X$, where rows of $X$ represent observations, and columns of $X$ represent variables, PCA creates the first PC, the linear combination of variables that can maximally explain the multivariate variability in $X$, using the eigenvector of the covariance matrix $X'X$ with the largest eigenvalue. The second PC is the eigenvector of $X'X$ with the second largest eigenvalue, which explains the maximum variability of $X$ subject to it being orthogonal to the first PC. In canonical analysis, canonical variables are chosen based on the ability of explaining the ratio of the between-population variance $B$ over the within-population variance $W$, so that in the first canonical variable the populations are maximally separated. Therefore, the first canonical variable is derived as the eigenvector of $W^{-1}B$ with the largest eigenvalue, the second canonical variable is chosen as the eigenvector of $W^{-1}B$ with the second largest eigenvalue, and so on.

Let $E$ be the matrix whose first column is the first eigenvector of $W^{-1}B$, the second column is the second eigenvector, and so on, and each eigenvector is scaled such that the within-population variance of the canonical variable is 1. A new data set $Y$ can be projected to the canonical space by

$$Y_p = Y_e E \quad (11)$$

where $Y_e$ is $Y$ with columns centered by subtracting their means and $Y_p$ is the projected data set.

Both PCA and canonical analysis project data to another space by linear transformation, but the objectives of the transformations are quite different. PCA orders the created variables according to the amount of the variability in the data the variables can explain, while canonical analysis orders the created variables according to the amount of the between-population variance over the within-population variance the variables can explain. Both PCA and canonical analysis can be used for feature reduction. With a limited number of created variables which is smaller than the dimension of the original data, PCA preserves the variability of the data and canonical analysis preserves the separation between the populations of the data. Because the separation between populations is compacted into a small number of variables, the populations of data will be maximally separated in the space formed by the first few canonical variables, and a standard classification algorithm can much more easily classify the samples.

In multivariate statistical analysis, the canonical analysis can be regarded as canonical correlation analysis [14], [19], [20] between the dependent variables and some dummy variables. The description above for the process of deriving the canonical variables is similar to Fisher’s linear discriminant analysis (LDA) [21], in which the canonical variables are known as discriminants. Applying the canonical analysis to our application, we categorize the normal chips as one population and the test escapes as the second population, and create the canonical variables by linear combinations of the test items.

IV. Test Methodology

Based on the feature engineering scheme discussed in Sections II and III, we propose to use two distinct sets of features. The first set of features are the residual vectors with measurement means as the estimated values, followed
by the transformation to the canonical space. The second set of features are the residual vectors which use predicted values from the learned spatial patterns as the estimated values, followed by the transformation to the canonical space.

The two sets of features can be utilized in two ways. First, each set of features is used as the input features for one classifier-determining if the chip under test belongs to the normal population or test escape population, and the classifiers together form a series of statistical tests. A second possible way of utilizing the features is to include all sets of features to form a single comprehensive statistical test. The exemplar test flows are demonstrated in this section.

![Fig. 3: Test flow with sequential statistical tests](image)

**A. Classifier**

In this paper we use the C-support vector classification (C-SVC) algorithm provided by the SVM library LIBSVM [22] as our classifier for separating test escapes and good chips. Given the fact that the two classes of samples are very imbalanced (i.e. the number of good chips is much greater than the number of test escapes) in a practical training set, one can set a much higher weight for the class of test escapes to force the algorithm to always find a model that correctly identifies escapes [23]. The guideline for our classification, however, is to screen out as many escapes as possible subject to the constraint of limiting the yield loss to a very small number (say, less than 0.001%). Based on this guideline, each class in the C-SVC is given the same weight in our experiment to allow a thorough search for a model with the maximum number of correctly identified escapes while minimizing the yield loss to a level very close to 0.

**B. Pre-test Analysis**

To start the proposed statistical tests, the canonical variables and the C-SVC models based on the search for the optimal combination of parameters [22] need to be generated based on a set of training chips, and the optimal C-SVC model needs to be selected based on a set of validation chips. The training/validation set of good chips should be sampled across several lots and wafers, and the distribution of measurements in each lot should be checked for uniformity to validate that the training/validation set indeed properly represents a good-chip population. Some field returns (or known test escapes that pass the test program) are also required for finding the canonical variables and the classification models. Section V will show that only a very small ratio of returns/known test escapes is required in order to find a canonical space for achieving significant runtime reduction while preserving the discriminating power between the test escapes and the normal population.

![Fig. 4: Test flow with a comprehensive statistical test](image)

**C. Test Application**

An exemplar test flow of the proposed statistical tests, applied to each wafer/lot, is shown in Fig. 3. These statistical tests are performed after all physical tests are executed, and serve as additional rejectors which reject some of the bad chips that escape all physical tests. While in this paper we suggest two specific statistical tests only, additional statistical tests can be developed and applied based on more new features.

A modification from Fig. 3 is shown in Fig. 4, in which the sequential statistical tests are replaced with one single comprehensive statistical test. With the application of canonical transform, the useful information in separating the test escapes and the normal population in all generated features
is incorporated into the comprehensive test. An experimental comparison on the two exemplar flows will be made in Section V.

Given any training set, it is possible that the manufacturing process drifts over time such that the training set is no longer sufficiently representative for the later data. Therefore, if the proposed statistical tests report an abnormally large number of test escapes for a wafer/lot under test, it could be due to such temporal process variation and thus the wafer/lot should be analyzed for the actual cause. If the wafer/lot is diagnosed as an outlier wafer/lot, we can conclude that the training set still effectively represents the population of good chips, and the test flow can continue to the next wafer/lot without a new training set. Otherwise, a new training set should be established and the corresponding canonical variables and C-SVC models should in turn be derived.

V. EXPERIMENTAL RESULT

To validate the proposed methods, the continue-on-fail production test data of a high volume commercial product was first preprocessed to remove confidential information while accurately preserving the information critical to the evaluation. The data set contains more than 1200 wafers with more than 200 parametric test measurements captured by the production test program for each die, and have more than 700 dies per wafer. In the following discussion we use $N$ to denote the number of parametric test measurements.

Since there is no actual test escape information in this data set, we first introduce how to emulate test escapes for our evaluation, and then validate our proposed methodology based on the data set.

A. Data Setup

In a test program, chips with measurements beyond the test limits are rejected as faulty chips, and those pass the test program but fail later at the system level or in the field (field returns) are test escapes. Without actual test escape information, we identified a set of bad chips as emulated test escapes which meet the following criterion: among the over 200 measurements, only one measurement did not fall within its test limits and its violation to the spec was marginal. We therefore hid this failing measurement which rejected these faulty chips - pretending that each of these bad chips still passed all physical tests in the test program and is treated as a test escape. To hide this measurement which failed a chip, we replaced its value by a normal value well within the test limits such that the resulting feature value (i.e. an element of the chip’s residual vector) is equal to the median of all good chips. After such manipulation, these faulty chips now have similar characteristics as test escapes: passing the complete test program, but with some intrinsic defects. We can then evaluate if the measurements of all-but-one test items which did not violate the test limits can expose those emulated test escapes in the proposed statistical tests.

The goal of our analysis is to screen test escapes based on the subtle differences in the non-failing measurements, so we only considered those faulty chips with a very small number of failing measurements and insignificant violations to the test specs as emulated test escapes.Faulty chips which fail many test items and/or have significant violations to the test specs are more likely to be catastrophic failures. Such catastrophic failures should have very revealing and differentiable features derived from the non-failing measurements, and including those catastrophic failures could result in an overly optimistic conclusion of the experiment. Therefore, we excluded faulty chips with more than one failing measurements or with only one failing measurement but its violation to the spec is significant from our analysis.

B. Sequential Rejectors

In pre-test analysis, test data from 200 wafers were used as the training set for finding the canonical variables and generating C-SVC models based on the search for the best combination parameters. Test data from another 300 wafers were used as the validation set for selecting the best C-SVC model with an acceptable level of yield loss.

For simplicity and clarity, we use the following notations for the different residual vectors:

- Feature Set $F_m$: Residual vectors derived using measurement means as the estimated values
- Feature Set $F_s$: Residual vectors derived using predicted values based on the learned spatial patterns as the estimated values

After the canonical space is found based on the training set, the residual vectors of the validation set are transformed into the canonical space by Eq. (11). For comparison, we also transform these residual vectors into the PC space by PCA. Fig. 5 illustrates the feature set $F_m$ transformed into the PC space and the canonical space respectively (with respect to the first three variables created in each of these two analyses). Fig. 6 shows feature set $F_s$ transformed into the PC space and the canonical space respectively. For better visualization, Fig. 5 and Fig. 6 show only a subset of good chips and emulated test escapes (they were randomly sampled from the validation set while maintaining the original ratio of good chips vs. test escapes). It is very clear that the test escapes are much more separable from the normal population in the canonical space than in the PC space for both $F_m$ and $F_s$.

![Fig. 5: The distributions of good chips/test escapes in different feature spaces of $F_m$.](image-url)
Fig. 6: The distributions of good chips/test escapes in different feature spaces of $F_s$.

Fig. 7 shows the relative operating characteristics (ROC) diagram of C-SVC’s performance based on $F_m$ and $F_s$ in the original space, the PC space, and the canonical space. The horizontal axis shows the yield loss rate and the vertical axis shows the test escape identification rate, which are the false positive rate and the true positive rate respectively in our classification problem. Using all features in the three spaces results in similar classification performances because the discriminating information is preserved in the linear transformation to a new space. When only three features are used for the purpose of feature reduction, using the first three canonical variables as input features results in significantly greater classification accuracy than using the first three PCs, which explain only 25% and 21% of the variability in $F_m$ and $F_s$ respectively.

Table I shows the ratio of test escapes identified by C-SVC in the $N$-dimensional original feature space, 3-dimensional PC space, and 3-dimensional canonical space, with a limit of 0.001% yield loss for all three cases. The first column shows the ratio of test escapes identified only by $F_m$ and not detectable by $F_s$. The second column shows the ratio of test escapes identified only by $F_s$ and not detectable by $F_m$, and the third column shows the ratio of the union of the test escapes identified by $F_m$ and $F_s$. C-SVC in the 3-dimensional PC space cannot identify any of the test escapes given the very low yield loss rate limit, while C-SVC in the 3-dimensional canonical space achieves a lower but still significant ratio of identified test escapes than C-SVC with all the features in the original space.

For a fixed yield loss budget (0.001% in this experiment), the learned C-SVC models based on different sets of features can identify unique sets of escapes. In the canonical space, there are 11.0% of the escapes that can be identified based on both $F_m$ and $F_s$, while the classifications based on $F_m$ and $F_s$ identify unique sets of 37.2% and 9.1% of the escapes, respectively.

To better visualize a feature set’s ability of uniquely identifying test escapes, Fig. 8 shows the distribution of identified test escapes in the canonical space of $F_m$. It is clear that $F_s$ reveals some test escapes that are close to the normal population in the canonical space derived from $F_m$, which C-SVC in this space cannot correctly classify without incurring additional yield loss. Therefore, it is important to incorporate both sets of features to screen test escapes more effectively.

### C. Comprehensive Test

In addition to applying canonical transform to $F_m$ and $F_s$ separately, we can apply canonical transform to $F_m$ and $F_s$ together, considering them as a single feature set for the classification problem. In this case, the proposed framework becomes even more extensible as one can generate many possible features and input them all into the canonical transform without knowing which sets of features are more suitable for which product. A general feature set can be developed and applied to different products efficiently since canonical transform automatically generates the most discriminating features out of all possible features for each product. The test flow is shown in Fig. 4 in Section IV.
Fig. 8: The distributions of good chips and test escapes detected by the two sets of features in the canonical space of $F_m$.

Fig. 9 shows the distribution of good chips and test escapes in the canonical space derived from $F_m \cup F_s$, i.e. a canonical variable is a linear combination of all features in $F_m$ and all features in $F_s$. There is a clear separation between the good chips and the test escapes. Fig. 10 shows the ROC curves of using 3 canonical variables derived from $F_m \cup F_s$, as the input features for C-SVC. When the yield loss rate is very low (< 0.001%), using the canonical variables derived from $F_m \cup F_s$ identifies 64.6% of the test escapes, which is higher than using the canonical variables derived from $F_m$ or $F_s$ alone. Compared with the results in Table I, using canonical variables derived from $F_m \cup F_s$ identifies more test escapes than the union of the test escapes identified using the canonical variables derived from $F_m$ and the canonical variables derived from $F_s$. In other words, using a single comprehensive test derived from all generated features achieves greater accuracy than using a sequence of rejectors, each of which is derived from a unique set of features.

D. Test Application

The testing set of our data includes more than 700 wafers containing more than 500K chips. The percentage of detected test escapes based on the testing set, given a yield loss budget of 0.001%, is shown in Table II. The classification performance is greatly enhanced in the canonical space than that in the PC space. Using the canonical variables derived from $F_m \cup F_s$ also results in much better accuracy than using the canonical variables derived from $F_m$ or $F_s$ alone. Note that in the original space there are $N$ dimensions in $F_m$ and $F_s$, $2N$ dimensions in $F_m \cup F_s$, where $N$ is the number of test measurements. TABLE II: Percentage of Test Escapes Detected by C-SVC in Three Different Spaces Based on the Testing Set

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>$F_m$</th>
<th>$F_s$</th>
<th>$F_m \cup F_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-SVC in original space</td>
<td>64.4%</td>
<td>23.6%</td>
<td>67.1%</td>
</tr>
<tr>
<td>C-SVC in 3-D PC space</td>
<td>0%</td>
<td>0%</td>
<td>0.07%</td>
</tr>
<tr>
<td>C-SVC in 3-D canonical space</td>
<td>43.4%</td>
<td>17.1%</td>
<td>61.9%</td>
</tr>
</tbody>
</table>

The following two tables show the runtimes executed on an Intel Xeon Quad-core 3.6GHz system. The data is based on the comprehensive test, in which the canonical variables are derived from $F_m \cup F_s$.

The runtime for canonical transform and PCA to derive the transform matrix based on the training set and to apply the transform on the testing set is shown in Table III. While it takes slightly longer to derive the transform matrix for canonical transform during the training process, the time for applying the transform is very close for both transforms.

TABLE III: Runtime of Deriving/Applying Transform Matrix for Canonical Transform and PCA Based on Training/Testing Set

<table>
<thead>
<tr>
<th>Transform</th>
<th>Canonical Transform</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Derivation</td>
<td>5.55 s</td>
<td>3.56 s</td>
</tr>
<tr>
<td>Application</td>
<td>0.75 s</td>
<td>0.78 s</td>
</tr>
</tbody>
</table>
Table IV shows the runtime of training the C-SVC model based on the training set and applying the C-SVC model to the testing set. Training in the canonical space is much easier because most of the separating power in the data is compacted into the 3 input features for C-SVC. A runtime reduction of 63X and 29X is achieved for training and applying the model in the canonical space. In application of the proposed statistical test to the test flow, it takes 0.75s + 8.41s to transform the features and apply the model to more than 700 wafers, resulting in less than 0.013s additional test time per wafer.

TABLE IV: Runtime of Training/Applying C-SVC Model in Three Different Spaces Based on Training/Testing Set

<table>
<thead>
<tr>
<th>Feature Space</th>
<th>Training</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>2N-D Original</td>
<td>31.27 s</td>
<td>242.16 s</td>
</tr>
<tr>
<td>3-D Canonical</td>
<td>0.50 s</td>
<td>8.41 s</td>
</tr>
<tr>
<td>3-D PC</td>
<td>3.42 s</td>
<td>74.87 s</td>
</tr>
</tbody>
</table>

E. Another Experimental Scenario

The emulated data set based on the previous description contains 3500PPM of test escapes, a good fraction of which are detected by a very small number of test items. In order to capture a more realistic scenario for which each test item detects only a small number of emulated test escapes, we removed some escapes from the original emulated test escape population (of 3500PPM) so that each test item only detects a limited number of escapes in the resulting test escape population. Specifically, we identified those test items that hiding each of them would result in greater than 50PPM in the original test escape population. We then removed those emulated escapes detected by these test items, resulting in a reduced emulated test escape population of 600PPM.

Based on the data set with a reduced number of test escapes, the ROC curves for C-SVC with $F_m \cup F_s$ as the input features are shown in Fig. 11. Note that in this case, C-SVC in the canonical space, even with only the first 3 dimensions, achieves greater classification accuracy than C-SVC in the original space with all 2N features. For a yield loss rate limited to 0.001%, Table V shows the test escape identification rate and the corresponding runtime for applying the model. In this data set, effectively compacting the separation between classes into the very few dimensions allows more effective and efficient classification for C-SVC than that in the original space with much more dimensions.

Fig. 12 shows the ROC diagram of classification based on 3 canonical variables derived from $F_m$, $F_s$, and $F_m \cup F_s$. Given the yield loss rate limited to 0.001%, the ratio of identified test escapes are 14.6%, 0.7%, and 16.3%, respectively. While the identification rate drops for the three cases compared with that in the original data set, classification based on the canonical variables derived from $F_m \cup F_s$ still achieves better accuracy than using the canonical variables derived from $F_m$ or $F_s$ alone, and the identification rate of test escapes is still significant under the constraint of a close-to-zero limit on yield loss rate.

VI. CONCLUSIONS

Through feature engineering, we propose two sets of features to characterize the health of chips. We demonstrate that statistical tests based on each set of features could uniquely identify some test escapes that the other set of features cannot reveal. As adding more features may reveal more test escapes, we further propose to transform these features into a canonical space for feature reduction. Classification performed by the statistical tests on the reduced dimensions in canonical space achieves 29X runtime reduction while achieving a significantly
higher accuracy than PCA in our experiment. We can expect further improvement if more types of features are added into the framework, followed by feature reduction through canonical analysis. Using our data set with emulated test escapes, we demonstrated that classification in a 3-dimensional canonical space can achieve greater accuracy than that in the original space with 200+ dimensions.

While C-SVC is used as the classification engine to evaluate various aspects of feature engineering proposed in the paper, other classifiers can also be used. The scheme to utilize statistical tests containing the proposed features is flexible and can be easily extended to accommodate more types of statistical tests.

ACKNOWLEDGMENTS

We would like to acknowledge the Semiconductor Research Corporation (SRC) and the Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan for their support on this work, and the input by Dr. Chun-Lung Hsu of ITRI throughout the discussions.

REFERENCES