On Proving the Efficiency of Alternative RF Tests

Nathan Kupp*, Haralampos Stratigopoulos†, Petros Drineas‡, and Yiorgos Makris§
*Department of Electrical Engineering, Yale University, New Haven, CT 06511
†TIMA Laboratory (CNRS-Grenoble INP-UJF), 46 Av. Felix Viallet, 38031 Grenoble, France
‡Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
§Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080

Abstract—**The deployment of alternative, low-cost RF test methods in industry has been, to date, rather limited. This is due to the potentially impaired ability to identify device pass/fail labels when departing from traditional specification test. By relying on alternative tests, pass/fail labels must be derived indirectly through new test limits defined for the alternative tests, which may incur error in the form of test escapes or yield loss. Clearly, estimating these test metrics as early as possible in the test development process is key to the success of an alternative test approach. In this work, we employ a test metrics estimation technique based on non-parametric kernel density estimation to obtain such early estimates, and, for the first time, demonstrate a real-world case study of test metric estimation efficiency at parts-per-million levels. To achieve this, we employ a set of more than 1 million RF devices fabricated by Texas Instruments, which have been tested with both traditional specification tests as well as alternative, low-cost On-chip RF Built-in Tests, or "ORBiTs".**

I. Introduction

In the post-silicon production flow, every integrated circuit is thoroughly tested before it is shipped to the user, in order to guarantee that it meets the original design specifications. Testing targets the detection of defects that are due to the various sources of imperfection in the fabrication technology. Defects can range from catastrophic to parametric. The former lead to a complete malfunction of the IC and, typically, can be detected by simple tests. The latter are caused by excessive process variations that may bring some or all of the specifications outside the allowable limits. Parametric defects are considerably harder to detect. For the case of RF circuits, the current practice is to measure directly the specified performances that are promised in the data sheet. Although this approach is highly accurate, it comes at the expense of a very high cost, which can amount up to 50% of the overall production cost according to anecdotal evidence. Given that RF circuits typically occupy less than 5% of the die area, it is unsurprising that the reduction of RF test cost is an area of focus and innovation for the semiconductor industry [1], [2].

The high cost of RF test is due to the expensive and sophisticated automated test equipment that is required, on one hand, and due to the lengthy test times that result from a sequential measurement approach, on the other hand. Recently, there has been an intensified effort to develop alternative test approaches that relax the requirements on test equipment and/or reduce the associated test times. Among others, the built-in test solution is perhaps the most promising and advantageous [3]–[6]. It relies on extracting on-chip digital, DC or low-frequency test signatures that carry RF information. Thereafter, these test signatures can be transported off-chip and processed by an inexpensive tester with minimum requirements.

Despite the number of alternative RF test approaches which have been proposed to date, the industry seems reluctant to replace the current test approach. The primary reason is the lack of automated tools for evaluating a new test approach fast and early at the design and test development phases, before moving to production test. It may be easy to estimate the area overhead incurred by a built-in test solution and to study to what degree it degrades the device performances, yet it is extremely difficult to estimate the incurred indirect costs, that is, the resulting test errors. A new test approach should reduce test cost without sacrificing test accuracy, that is, it should result in minimum test escape T_E (e.g. faulty devices that pass the test) and yield loss Y_L (e.g. functional devices that fail the test).

This paper presents a case study of test metrics estimation. Specifically, the aim is to prove the equivalence of low-cost On-chip RF Built-in Tests (ORBiTs) to the traditional RF specification tests, based solely on a small data set obtained at the onset of production. Our initial judgement is confirmed on a much larger data set containing more than 1 million Bluetooth/Wireless LAN devices fabricated by Texas Instruments. In Section II, we explain the challenges of providing early and accurate test metric estimates and we provide an overview of the case study. In Section III, we discuss the problem of setting limits on ORBiTs. In Section IV, we detail the test metrics estimation method which we leverage to obtain accurate early estimates at low cost. In Section V, we define a feature selection method which we use to focus our analysis on subsets of ORBiTs. Finally, in Section VI, we provide experimental validation of the test metric estimation method.

II. Proving the efficiency of alternative tests

Consider, for example, an arbitrary candidate alternative test system $f(\cdot)$ shown in Figure 1, which operates on the device under test D_i and maps a collected set of alternative measurements X_i to device labels y_i, identifying each device as passing or failing. Such a system will realize a classification efficiency in terms of T_E and Y_L, which are simply measures on the incorrectly labeled devices.

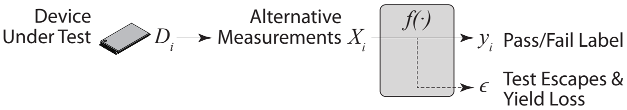

Fig. 1. Candidate alternative test system

978-1-4577-1400-9/11/$26.00 ©2011 IEEE

The most readily obvious way to characterize the alternative test system under consideration and obtain accurate, parts-per-million (ppm) test metric estimates is to take a very large set of fabricated devices, say 1 million, and apply the alternative test system to each device, recording test metrics on each, as shown in Figure 2. However, this is not a sustainable practice for evaluating candidate alternative test systems. Indeed, through this analysis we may ultimately conclude that the alternative test system ends up having unacceptably large test metrics, in which case we have inadvertently wasted a great deal of test resources and test time.

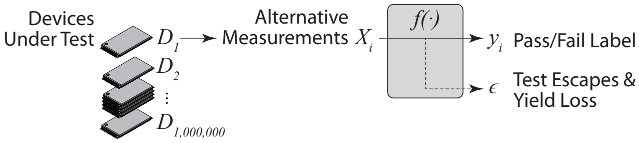

Fig. 2. Obtaining parts-per-million test metric estimates

Furthermore, the estimation of test escapes and yield loss in a simulation environment is difficult because these events are "rare" and cannot quickly be reproduced with high fidelity. Typically, a robust design will result in a very small defect level (e.g. percentage of devices that are faulty), on the order of a few thousands of faulty ppm. Thus, the test escape rate is typically in the order of a few hundreds ppm, which corresponds to a probability of around 10^{-5}. Similarly, a decent test will fail a small fraction of the functional devices, which corresponds to a yield loss probability of a similar order. This implies that millions of Monte Carlo simulations are needed to estimate such low probabilities with the required accuracy, which is clearly computationally infeasible. Most often, practitioners examine the behavior of tests for a few corner cases, but this approach does not reveal the full truth since a process design kit has dozens of parameters which implies a intractable number of corner cases.

In this work, we employ a general technique for obtaining ppm test metric estimates, originally developed in [7], and we examine for the first time its potential on a real-world case study. This technique is able to elegantly achieve the objective of providing such accurate estimates, as in Figure 2, while reducing the required investment, that is, without the extreme cost associated with having to consider millions of fabricated devices. It is based on the statistical methodology of non-parametric kernel density estimation (NKDE), as shown in Figure 3. The underlying idea is to rely on a small set of representative devices to estimate the joint non-parametric probability density function of specified performances and alternative tests. Thereafter, the estimated density is sampled to generate a large synthetic set of device instances from which one can readily compute test metrics using relative frequencies.

Moreover, we are able to provide a case study demonstrating equivalence of our proposed system of Figure 3 and the true ppm metrics obtained via explicitly testing 1 million devices in Figure 2. The case study concerns a Bluetooth/Wireless LAN device from Texas Instruments for which a set of ORBiTs [6] are developed to replace the costly standard specification tests. We have at hand measured data from more than one million device instances. These measured data include the specified performances and the ORBiTs. Thus, we have sufficient information to compute the true test metrics resulting from the replacement of specification tests with the lower-cost ORBiTs alternatives.

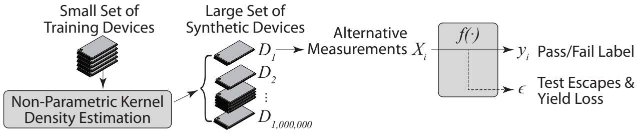
Small Set of Training Devices
Non-Parametric Kernel Density Estimation
Large Set of Synthetic Devices
D_1
D_2
⋮
$D_{1,000,000}$
Alternative Measurements X_i
$f(\cdot)$
y_i Pass/Fail Label
ϵ Test Escapes & Yield Loss

Fig. 3. Low-cost method for obtaining parts-per-million test metric estimates

The ORBiTs have been proven to be generally very efficient in such replacements, but this knowledge was acquired only after measuring millions of RF device instances with the dedicated built-in test circuitry. In this paper, we try to answer the following question: Is it possible to estimate values of the test metrics close to true ones while employing in the analysis a small set of RF devices that we obtain at the onset of production? In this case, we will be able to decide on the efficiency of the ORBiTs early in the process without having to wait for a large volume of silicon data to reach a safe conclusion. This type of proactive analysis is very important in cases where the alternative tests are found later on to be inefficient. It allows to convince test engineers about the efficiency of an approach, to identify shortcomings and come up with remedies for refining an approach, or abandon an approach altogether if it is deemed not to be equivalent to the standard specification test approach.

III. Setting test limits on ORBiTs

Before explaining the test metric estimation approach of Figure 3, it is necessary to define the alternative test system $f(\cdot)$ being employed and the means of mapping ORBiT measurements to pass/fail labels, e.g. via setting test limits. The ORBiTs are internal tests designed to replace expensive traditional specification tests by self-testing the device with hardware available on-die. Specifically, the ORBiTs are targeted at two RF cores: a Bluetooth radio and wireless LAN radio, which are components in a much larger device containing an ARM core. The ARM core is used in conjunction with on-die test structures to compute test outcomes; thus, the ORBiTs are entirely internal to the device, and only the test results are reported externally. Further information about the ORBiTs employed can be found in [6].

Consider a specification performance P which we wish to replace with the ORBiT set $X = \{f_1, f_2, \ldots, f_{d_0}\}$. If a subset of X were to contain all the information necessary to correctly label each device as passing or failing with the same efficiency as P, it would be a trivial task to set test limits and incur no additional test escapes or yield loss by reliance on the system $f(\cdot)$. In reality, any ORBiT replacement for P is likely to be imperfectly correlated, and therefore realize slightly increased error metrics. Ultimately, the resultant test metrics become a function of the test limits imposed on the ORBiTs. Therefore, before investigating the technique for estimating error metrics, we must fully specify the test limits and alternative test system $f(\cdot)$ we are using. However, we remind the reader that the particular choice of test limit-setting technique is largely auxiliary to the primary objective of this work, in that we are interested in estimating test metrics.

An intuitive approach to defining $f(\cdot)$ is to carefully set a limit on each individual ORBiT, as one would do with traditional specification tests. This would result in a hyper-rectangle acceptance area in the space of ORBiTs, as depicted for the specification test space in Figure 4. However, one has to take into consideration that ORBiTs are alternative tests that are not specified in the data sheet. To this end, one has to set the limits on ORBiTs such that the faulty devices are separated from the functional ones, where the labels are assigned according to the actual specification test limits. Since the ORBiTs space is a complex translation of the specification test space, the separation boundary becomes highly irregular and non-linear and, thereby, a hyper-rectangle would be a crude approximation that inadvertently would give rise to test errors. Indeed, the passing subspace may even be non-convex in the ORBiT space or imperfectly defined if the ORBiTs do not capture all the same information latent in the specification tests.

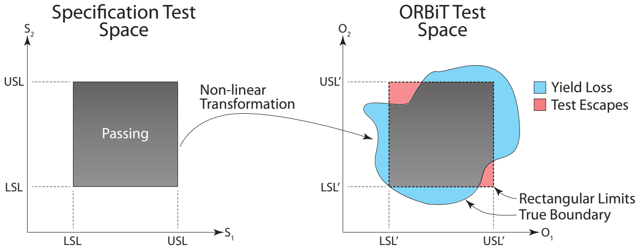

Fig. 4. Specification boundary translation

In this work, we improve on the limit-setting approach by implicitly learning the complex boundary which defines passing and failing regions in the ORBiT space [8], instead of explicitly defining it as a hyper-rectangle. By relying on machine learning, we are readily able to capture the non-linear, complex and possibly non-convex nature of the limits in ORBiT test space and avoid the test escape and yield loss shown in Figure 4. Specifically, we train a support vector machine (SVM) [9] to learn the boundary. As will be described in greater detail in section VI-E, the training phase employs an information-rich synthetic set of device instances that is generated through statistical simulation. This set comprises marginal instances whose footprints in the ORBiT space cover the areas around the true separation boundary [10] and, thereby, they allow a good approximation of this boundary.

IV. Test metrics estimation method

With the machine learning approach to setting test limits defined in section III, we can readily proceed to evaluating the test error metrics introduced. As noted in the Introduction, test metrics can be on the order of hundreds of ppm, which is likely beyond the capacity of a small device data set to capture. However, this reliance on small data sets for validation is understandable given the very costly alternative of validating on a data set with hundreds of thousands or millions of devices, which would require *both* ORBiT and traditional specification tests to be collected on every device in the validation set. Clearly this is a non-trivial cost overhead simply to prove the efficacy of the chosen test limits.

It is in this context that we introduce a novel methodology originally proposed in [7] to obtain test metric estimates with ppm accuracy, while side-stepping the cost associated with exhaustively testing millions of devices. This technique, based on NKDE, permits dramatically enriching the validation set with synthetic device instances reflective of the true device population. With this large synthetic device set in hand, we are able to produce test metric estimates using relative frequencies. In particular, if we denote by N, N_{fp}, and N_{gf} the size of the synthetic device set, the number of faulty devices in this set that pass the ORBiTs, and the number of functional devices in this set that fail the ORBiTs, respectively, then $T_E \approx N_{fp}/N$ and $Y_L \approx N_{gf}/N$.

NKDE relies on a small Monte Carlo run (e.g. on the order of a few thousands devices) to generate a synthetic device sample with population statistics nearly identical to the 10^6-order population we are unable to simulate. The underlying idea is to estimate the joint probability density function of ORBiTs and specification tests based on the small Monte Carlo run. Instead of assuming a specific parametric form for the probability density function (e.g. Gaussian), NKDE makes no *a priori* assumptions and allows the available simulation data to speak for themselves.

Formally, let x denote the vector comprising the ORBiTs and specification tests and let x_i, $i = 1, \ldots, n$, denote n available observations of x. We first position an Epanechnikov kernel centered on each observation defined by

$$K_e(t) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-t^Tt) & \text{If } t^Tt < 1 \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

where $c_d = 2\pi^{d/2}/(d \cdot \Gamma(d/2))$ is the volume of the the unit d-dimensional sphere, and d is the dimensionality of x. Then, we introduce a non-parametric estimate $\tilde{f}(x)$ of the true density $f(x)$ as the normalized sum of all observation-centered kernel functions

$$\tilde{f}(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K_e\left(\frac{1}{h}(x - x_i)\right) \tag{2}$$

where h is a smoothing parameter known as bandwidth. It can be shown that $\tilde{f}(x)$ converges to $f(x)$ as $n \to \infty$ for a proper choice of h. The interested reader is referred to [7], [11] for an in-depth discussion.

Thereafter, we can repeatedly sample (e.g. simulate) $\tilde{f}(x)$ to obtain new observations of x, thus generating a synthetic population of devices of arbitrary size. Millions of samples can be generated in a few minutes using a standard desktop PC. This is similar to generating random samples from a Gaussian distribution in MATLAB, yet herein we derive the original probability density function without assuming a specific parametric form.

V. Feature Selection

Often when dealing with early characterization data sets, a large number of measurements are available which are later pruned for the final test set. This provides a wealth of data, but also presents a case of the well-known "curse of dimensionality", the law that by adding dimensions, one exponentially increases data sparsity. This data sparsity can cause learning algorithms to have high-variance classification boundaries and poor generalization capability. Consider, for example, an arbitrary ORBiT subspace as illustrated in Figure 5, containing only a single passing device A and failing device B. Considering that only these two devices are available for learning the boundary, our classifier might simply allocate a straight line boundary separating these two points (shown in the figure as the dotted line), where the true boundary is far more complex. Thus, classification learning algorithms are often trained on some subset of the available features rather than all of the available features, in order to concentrate the data in a subspace and reduce the variance of the classification boundary. A variety of methods exist to perform this pruning, typically conditioned on the empirical classification error in a hold-out set or some similar error measure in order to determine an appropriate subset of features to retain.

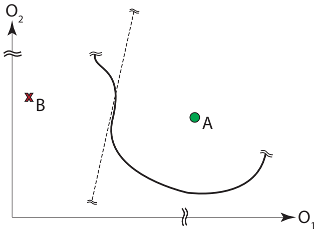

Fig. 5. Allocation of boundaries in sparsely populated spaces

A key component of our analysis was to perform feature selection on the very high dimensional space of available ORBiTs, projecting device test signatures into a more reasonable low dimensional subspace. Generally, feature selection is a difficult problem, since the number of possible subsets of features is $2^{d_0} - 1$, where d_0 is the cardinality of the complete feature set. With even a moderate number of features, exhaustive search of the feature space is completely untenable. An excellent review of various approaches to feature selection is given in [12], and within the analog/RF test community several heuristic methods have already been employed, such as genetic algorithms [13].

In this work we make use of a supervised feature selection method known as Laplacian score feature selection (LSFS) [14] to rank ORBiTs and subsequently reduce dimensionality. LSFS is a supervised feature selection technique which ranks features according to their locality preserving power. It considers the passing and failing classes to define two neighborhoods of locality in feature space. Formally, let

$$X_i = [f_1^{(i)}, f_2^{(i)}, \ldots, f_{d_0}^{(i)}] \tag{3}$$

denote the pattern of device instance i, $i = 1, \ldots, n$, where n is the total number of devices and d_0 is the total number of ORBiTs measured on these n devices. We will create a graph where the i-th node corresponds to the i-th device instance. Within the graph, we connect nodes i and j if device instances i and j have the same class (passing or failing). We then consider the similarity measure

$$S_{ij} = \begin{cases} e^{-\frac{||X_i - X_j||^2}{t}} & \text{Nodes } i, j \text{ from same class} \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

where t is a suitable constant[1]. For each feature $f_r, r = 1, \ldots, d_0$, we calculate the Laplacian score

$$L_r := \frac{\sum\limits_{i,j}(f_r^{(i)} - f_r^{(j)})^2 S_{ij}}{\sigma^2_{f_r}} \tag{5}$$

using the samples $f_r^{(1)}, f_r^{(2)}, \ldots, f_r^{(n)}$, where $\sigma^2_{f_r}$ denotes the variance of f_r. We rank order the features according to their Laplacian score L_r

$$L^{(1)} \leq L^{(2)} \leq \ldots \leq L^{(d_0)} \tag{6}$$

such that

$$L^{(1)} = \min\{L_1, \ldots, L_{d_0}\} \tag{7}$$

$$L^{(2)} = \min\Big(\{L_1, \ldots, L_{d_0}\} - L^{(1)}\Big) \tag{8}$$

$$\vdots$$

$$L^{(d_0)} = \min\Big(\{L_1, \ldots, L_{d_0}\} - \{L^{(1)}, \ldots, L^{(r-1)}\}\Big) \tag{9}$$

Finally, we define a threshold τ_L on the Laplacian scores and retain features that have $L^{(i)} \leq \tau_L$. Let $d_0^{'}$ be the number of retained features which are denoted by $f_1^{'}, \ldots, f_{d_0^{'}}^{'}$.

VI. Experimental results

A. Data set

To confirm the efficiency of our approach in providing early estimates of test metrics, we employed a Texas Instruments data set from a total of more than 1.1 million devices. The devices are collected from 176 wafers and each wafer has between 6,000 and 7,000 devices. For each device, the data set contains the ORBiT measurements and the specified performances in the data sheet. Specifically, there are 739 ORBiT measurements $\{f_1, \ldots, f_{739}\}$ and 367 performances $\{P_1, \ldots, P_{367}\}$.

Some ORBiT measurements and performances are discrete-valued. The test metrics estimation method discussed in Section IV is defined only for continuous variables. Therefore,

[1] Generally, we have found $t = d_0$ to be an appropriate choice.

in our analysis we considered only the continuous ORBiT measurements and performances, which number 249 and 264, respectively. Formally, let unique(X_i), unique(P_j) denote the number of unique values observed across all devices in hand for the i-th ORBiT measurement and j-th performance, respectively. We consider only ORBiT measurements and performances with more than 100 unique values, that is, we define a threshold $\tau_D = 100$ and we retain the ORBiT measurements and performances that satisfy

$$\text{unique}(X_i) \leq \tau_D \tag{10}$$

$$\text{unique}(P_j) \leq \tau_D \tag{11}$$

B. Objective of the experiment

Recall that our objective is to predict the expected test escape T_E and yield loss Y_L test metrics if we replace the specification tests under consideration by an ORBiT subset. Consider $T_E^{(i)}$ the predicted test escape in the case where the specification test targeting performance P_i is replaced by an ORBiT subset. Then, the global test escape rate satisfies the following inequalities, measured in ppm

$$T_E \leq T_E^{(1)} + T_E^{(2)} + \ldots + T_E^{(367)} \tag{12}$$

$$\max_i \{T_E^{(i)}\} \leq T_E \tag{13}$$

Thus, if we can afford $T_E \leq \lambda$, using Equation (13) we can conclude that the ORBiTs are inappropriate if $\max_i\{T_E^{(i)}\} > \lambda$. Conversely, if $\sum_i T_E^{(i)} \leq \lambda$, then we can certainly use the ORBiTs. In fact, $\sum_i T_E^{(i)}$ represents a pessimistic upper bound for the T_E.

In our analysis, we focused on replacing the single most sensitive specification test, that is, the test that corresponds to most commonly failing performance across all wafers. We denote this performance by P. To predict the test metrics, we only use devices from the first wafer and we employ the KNDE technique to generate 1 million additional synthetic devices, in order to achieve ppm levels of accuracy, as illustrated in Figure 3. We emphasize that our use of a single wafer is purely to demonstrate the efficacy of the method in extremely challenging circumstances; in reality this sample may include an arbitrarily large training sample.

C. Removing Outliers

From the first training wafer, we remove outliers via a "defect filter", for two reasons. First, we do not wish outliers with non-statistical signatures to have leverage over the feature selection process; the retained features should excel at discerning the more difficult parametric fails rather than the relatively easy-to-detect catastrophic fails. Second, the test metrics estimation method itself relies on estimating a probability density function, thus we should avoid using outliers for this purpose since they are non-statistical in nature and are not generated by the same probability distribution which assumes only process variations. To remove such outliers from the first training wafer, let s_l, s_u be the lower and upper specifications for P. Then a device instance i will be removed if its performance $P^{(i)}$ satisfies

$$P^{(i)} > \kappa_u \cdot s_u \tag{14}$$

$$P^{(i)} < \kappa_l \cdot s_l \tag{15}$$

where κ_u and κ_l are appropriate user-defined constants. Herein, we employ $\kappa_u = \kappa_l = 3$. This results in the removal of approximately 3% of the device instances from the first wafer training set. Note that this step is necessary only for the training set, and subsequent outlier fails are not removed in this fashion. Moreover, the proposed methodology is not particularly sensitive to the choice of κ_u and κ_l as these limits only serve to remove extreme outliers.

D. Reducing the dimensionality of ORBiTs

As discussed in section V, fitting a classifier boundary in a sparse, high dimensional space can be error-prone due to the consequent variance of the fitted class boundary. For this purpose, we employ LSFS to reduce the dimensionality of the problem. In particular, for each of the 249 ORBiTs, we compute and rank the Laplacian scores as shown in Figure 6. In this experiment, a threshold of $\tau_L = 0.01$ was chosen, which corresponds to retention of 7 ORBiTs. It should be stressed that a lower dimensionality also maximizes the efficacy of NKDE technique which is also vulnerable to the "curse of dimensionality".

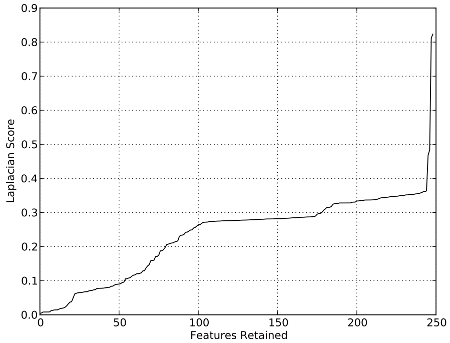

Fig. 6. Laplacian Scores

E. Information-rich training set

It turns out that even in relatively densely-populated spaces, classifier performance can benefit by further increases in data density. Specifically, it is not advisable to attempt to directly fit a classification boundary to a severely unbalanced population, as the classifier tends to always label subsequent instances as the dominant class after training.

To combat this effect and improve classifier performance by increasing data density in the training set, we employed non-parametric density estimation to generate synthetic training instances. To do this, we fit the joint probability density function of vector $x = [f_1^{'}, \ldots, f_{d_7'}^{'}, P]$ using the instances from the first wafer. We sample the empirical probability density function to generate an information-rich training set that has a more balanced population of good, faulty, and critical devices across the decision boundary in a similar fashion to the approach taken in [10].

F. Summary and Results

Assembling the preceding steps, we arrive at the complete analysis approach shown in Figure 7. The training set is employed to train the SVM classifier to assign limits on the 7 ORBiTs in the form of a hyper-surface boundary, as shown in Figure 4. The limits are used to obtain the ground truth test escape and yield loss values for each wafer, denoted by T_E and Y_L, respectively. These values are averaged to obtain the ground truth ppm test escape and yield loss measured over the complete device population in hand, denoted by $\overline{T}_E$ and $\overline{Y}_L$, respectively. The same limits are used on the synthetic device set generated from the first wafer, in order to obtain early ppm estimates of the test escape yield loss, denoted by $\hat{T}_E$ and $\hat{Y}_L$, respectively.

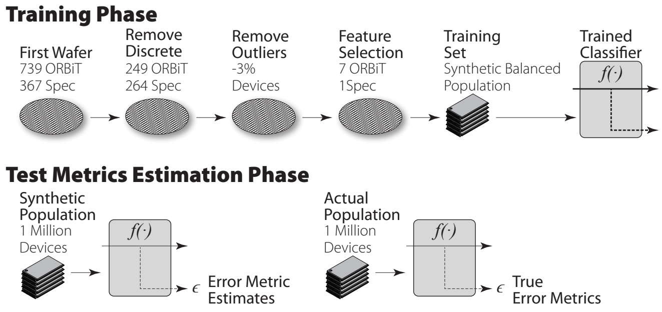

Fig. 7. Summary of experimental approach

The results are shown in Figure 8. As can be observed, test escape is slightly underestimated, and yield loss is very slightly overestimated. Specifically, the true values are $\overline{T}_E = 0.7286\%$ and $\overline{Y}_L = 4.387\%$, whereas the early estimates are $\hat{T}_E = 0.4302\%$ and $\hat{Y}_L = 4.401\%$, that is, a difference of $\Delta T_E = 0.2984\%$ and $\Delta Y_L = -0.014\%$. We remind that the objective of the paper is not to propose an alternative test technique, but to evaluate a proposed alternative test technique at an early phase. Moreover, we evaluated the scenario where a subset of ORBiTs replaces the most sensitive specification test, and not the general case where the complete suite of ORBiTs is used to replace irrespectively all specification tests.

VII. Conclusion

In this work we presented a method for providing accurate, parts-per-million estimates of test metrics without incurring the cost associated with simulating or testing millions of devices. A comparatively small set of RF devices from a single wafer tested at the onset of production coupled with the proposed NKDE-based sampling are used to generate one million synthetic device samples, on which we are able to evaluate test escape and yield loss test metric estimates. Furthermore, we have demonstrated our test metric estimates to be very close to the true values measured on more than one million devices from Texas Instruments.

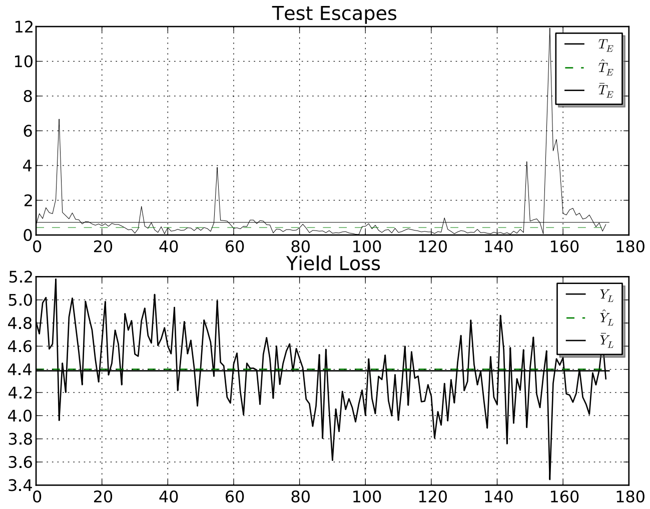

Fig. 8. Prediction across all wafers

References

[1] J. Ferrario, R. Wolf, S. Moss, and M. Slamani, "A low-cost test solution for wireless phone RFICs," *IEEE Communications Magazine*, vol. 41, no. 9, pp. 82–88, 2003.

[2] A. Abdennadher and S. A. Shaikh, "Practices in mixed-signal and RF IC testing," *IEEE Design & Test of Computers*, vol. 24, no. 4, pp. 332–339, 2007.

[3] S. Bhattacharya and A. Chatterjee, "A DFT approach for testing embedded systems using DC sensors," *IEEE Design & Test of Computers*, vol. 23, no. 6, pp. 464–475, 2006.

[4] L. Rolindez, S. Mir, J.-L. Carbonero, D. Goguet, and N. Chouba, "A stereo ΣΔ ADC architecture with embedded SNDR self-test," in *IEEE International Test Conference*, 2007, paper 32.1.

[5] M. Cimino, H. Lapuyade, Y. Deval, T. Taris, and J.-B. Bégueret, "Design of a 0.9V 2.45 GHz self-testable and reliability-enhanced CMOS LNA," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1187–1194, 2008.

[6] D. Mannath, D. Webster, V. Montano-Martinez, D. Cohen, S. Kush, T. Ganesan, and A. Sontakke, "Structural approach for built-in tests in RF devices," in *IEEE International Test Conference*, 2010, Paper 14.1.

[7] H.-G. Stratigopoulos, S. Mir, and A. Bounceur, "Evaluation of analog/RF test measurements at the design stage," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 4, pp. 582–590, Apr. 2009.

[8] H.-G. Stratigopoulos and Y. Makris, "Non-linear decision boundaries for testing analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, pp. 1760–1773, 2005.

[9] N. Cristianini and J. Shawe-Taylor, *Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge, 2000.

[10] H.-G. Stratigopoulos, S. Mir, and Y. Makris, "Enrichment of limited training sets in machine-learning-based analog/RF test," in *Design, Automation & Test in Europe*, 2009, pp. 1668–1673.

[11] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC, 1986.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[13] K. Deb, A. Pratap, A. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[14] X. He, D. Cai, and P. Niyogi, *Laplacian Score for Feature Selection*, In NIPS. MIT Press, 2005.