Data Learning Techniques and Methodology for Fmax Prediction *

Janine Chen[1], Li-C. Wang[1], Po-Hsien Chang[1]
Jing Zeng[2], Stanley Yu[2], Michael Mateja[2]

[1]Department of ECE, UC-Santa Barbara
[2]Advanced Micro Devices, Inc.

Abstract

The question of whether or not structural test measurements can be used to predict functional or system Fmax, has been studied for many years. This paper presents a data learning approach to study the question. Given Fmax values and structural delay measurements on a set of sample chips, we propose a method called conformity check whose goal is to select a subset of conformal samples such that a more reliable predictor can be built on. Our predictor consists of two models, a conformal model that decides on a given chip if its Fmax is predictable or not, and a prediction model that outputs the predicted Fmax based on results obtained from structural test measurements. We explain the data learning methodology and study various data learning techniques using frequency data collected on a high-performance microprocessor design.

1 Introduction

Binning is a common practice for dealing with performance variability. To assess performance variability, different types of measurement can be applied. For examples, common measurements include Iddq measurement, ring-oscillator based measurement, and test pattern based measurement with *frequency stepping*. In test pattern based measurement, the test patterns can be functional test patterns, transition-fault delay patterns, path-delay patterns, logic BIST patterns, and memory BIST patterns, etc.

Given results from two types of measurement, it is always interesting to study how they are correlated. The correlation is of particularly interest if one is seen as the golden method for measuring the performance, and the other as the potential lower-cost alternative. In order to use the lower-cost alternative for performance prediction, one desires to ensure a high correlation in between the two.

With system Fmax as the golden reference and structural Fmax based on path delay tests as the lower-cost alternative, the authors in [1] studied the correlation between the two using sample chips fabricated with 150nm technology. A linear relation were established: $Fmax_{system} = 4.428 + 0.7 * Fmax_{path}$ to demonstrate a high correlation.

With functional Fmax as the golden and structural Fmax as the lower-cost alternative, the authors in [2] studied the correlation based on five structural test pattern sets: complex transition pattern set, simple transition pattern set, Array BIST, complex path delay pattern set and simple path delay pattern set (where a complex pattern involved on-chip memory and a simple pattern did not). The samples were microprocessor parts running at above 1GHz. No linear relation was found in between any one of the five structural Fmax results and the functional Fmax. However, by dividing all frequency numbers into two bins, slow and fast, they were able to show, with respect to the binning scheme, that structural Fmax based on complex path delay patterns provided the best correlation to the functional Fmax.

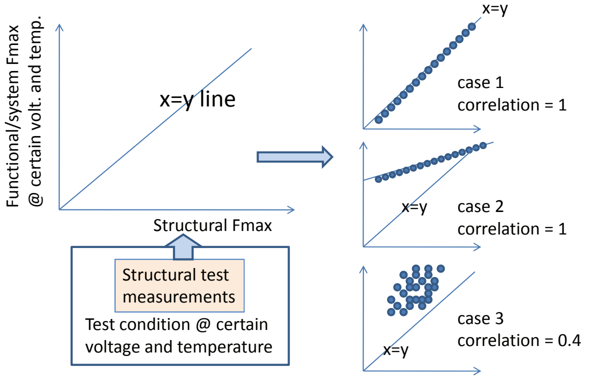

Figure 1. Correlating strcutural Fmax to functional or system Fmax - a two-dimensional perspective

In the two studies, one structural frequency, i.e. the maximum frequency passing all structural tests, is taken on each chip for correlating to the functional or system Fmax. The correlation is evaluated by checking how well the structural Fmax tracks the system or functional Fmax over a set of chips. The correlation can be checked with a two-dimensional plot where the two Fmax numbers on every

*This work is supported in part by National Science Foundation, Grant No. 0541192, Semiconductor Research Corporation, project task 1585.001

978-1-4244-4867-8/09/$25.00 ©2009 IEEE

chip are plotted as a point. Figure 1 illustrates such a plot. For examples, case 1 and case 2 in the figure show perfect correlation. Case 3 shows a correlation 0.4.

When using such a plot to study the correlation, one desires to see result similar to case 1 or case 2. If the result is like case 3, usually the effort for improving the correlation can be spent on finding better structural test measurements and/or better test conditions. For example, one may alter the structural test patterns by increasing or decreasing their switching activities and hope to find a better correlation.

In this work, our focus is not in finding a better test pattern set, in finding a better way to measure structural Fmax, or in finding a better test condition to improve the correlation using a two-dimensional plot. These approaches all involve altering the structural delay data.

In our approach, the data is given and fixed. The two-dimensional correlation perspective shown in Figure 1 is replaced with the multi-dimensional data learning perspective shown in Figure 2. And then, the objective is to learn an optimal Fmax predictor based on the dataset.

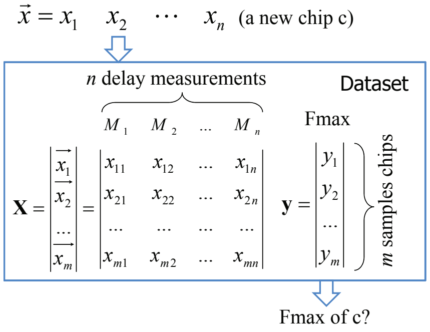
$\vec{x} = x_1 \quad x_2 \quad \cdots \quad x_n$ (a new chip c)
n delay measurements
Dataset
$M_1 \quad M_2 \quad \ldots \quad M_n$
Fmax
$\mathbf{X} = \begin{vmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \ldots \\ \vec{x}_m \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{vmatrix}$
$\mathbf{y} = \begin{vmatrix} y_1 \\ y_2 \\ \ldots \\ y_m \end{vmatrix}$
m samples chips
Fmax of c?

Figure 2. **Correlating structural test measurements to functional or system Fmax - a multi-dimensional data learning perspective**

In Figure 2, n structural frequencies are collected and utilized. For example, $M_1, \ldots, M_n$ can be the maximum passing frequencies on n flip-flops, based on a given set of transition fault patterns. Because on the structural side, we now have a vector of frequency numbers $(M_1, \ldots, M_n)$, its correlation to the system or functional Fmax can no longer be checked by using a simple plot as before.

The concept of ”correlation” in Figure 2 then turns into learning a prediction function P that can predict the Fmax based on the structural delay measurements. If a more reliable P can be learned from the dataset, we may say that the correlation is higher. The reliability of P can be characterized by the expected error it may induce on predicting a new chip c. For prediction, c is given with its structural measurement result as $\vec{x} = (x_1, \ldots, x_n)$. Then, we have ”predicted Fmax of $c = P(\mathbf{X}, \mathbf{y}, \vec{x})$.” Given the dataset $(\mathbf{X}, \mathbf{y})$, many prediction functions can be developed. Our goal is therefore to find an optimal P with minimal expected error.

1.1 Can a Fmax always be reliably predicted?

In general, the prediction accuracy on a chip c by P depends on the information content in the dataset $(\mathbf{X}, \mathbf{y})$. For example, if predicting the Fmax of c requires some information not contained in the dataset, there should be no reason that P is able to reliably predict the Fmax.

To illustrate this point, consider two simple approaches that might be used to implement P. The first approach is the *nearest neighbor* (NN) method [3]. In the NN method, $\vec{x}$ is compared to $\vec{x}_1, \ldots, \vec{x}_m$ and find the ”nearest neighbor” said $\vec{x}_i$. The ”nearest” can be defined based on the distance measure such as $||\vec{x} - \vec{x}_i||^2$. Hence, $\vec{x}_i$ can be the closest one in Euclidean distance to $\vec{x}$. Then, we let $P(\mathbf{X}, \mathbf{y}, \vec{x}) = y_i$, i.e. the Fmax of $\vec{x}$ is predicted as the Fmax of $\vec{x}_i$.

The NN method can be extended to k-NN method where k nearest neighbors are used. In the k-NN method, the predicted Fmax is the average of $y_{j_1}, \ldots, y_{j_k}$ where $\vec{x}_{j_1}, \ldots, \vec{x}_{j_k}$ are the k nearest neighbors to $\vec{x}$.

With a k-NN method, what happens if $\vec{x}$ is very far away from any one of the $\vec{x}_1, \ldots, \vec{x}_m$? Intuitively, this means that the delay measurement results collected on chip c are very different from the delay measurement results collected from any one of the m sample chips. In this case, using the Fmax numbers of the k nearest neighbors (still far way from $\vec{x}$) to predict c's Fmax could be unreliable.

The second approach is the *least square fit* (LSF) method [3] of a linear equation $f(\vec{z}) = w_1 z_1 + \cdots + w_n z_n + b = \vec{w} \cdot \vec{z} + b$. In LSF, the $n+1$ parameters $(\vec{w}, b)$ are found by minimizing the *square fitting error* which is calculated as: $SE = \sum_{i=1}^{m} ((\vec{w} \cdot \vec{x}_i + b) - y_i)^2$.

Again, consider the case that $\vec{x}$ is very different from any one of the $\vec{x}_1, \ldots, \vec{x}_m$. Since $f()$ is built by minimizing the fitting error on $(\mathbf{X}, \mathbf{y})$, it provides little guarantee to the error of prediction based on $\vec{x}$. In this case, the prediction error could be very large.

Because the prediction errors on different chips can vary significantly, it is important to evaluate such an error for each given chip before its predicted Fmax can be accepted. This idea is therefore at the center of our methodology.

The rest of the paper is organized as the following. Section 2 explains the main ideas behind the proposed methodology. Section 3 describes the first set of data targeting on functional Fmax. Section 4 presents the correlation results based on five learning techniques, the LSF [3], k-NN [3], ridge regression [3], Support Vector regression (SVR) [4], and Gaussian Process (GP) [5]. Section 5 presents a method called Conformity Check that tries to remove noisy samples from a dataset so that a better Fmax predictor can be built on. Section 6 discusses the second set of data targeting on system Fmax, and shows that the proposed methodology has similar effectiveness on this dataset as well. Section 7 explains the development and use of a conformal model based on one-class learning [4]. Section 8 concludes.

2 The Proposed Methodology

Correlation is a concept defined based on two variations. Consider why a high correlation may exist between a structural frequency variation and a functional or system Fmax variation. In Figure 2, the structural *frequency variation* is reflected in the diversity of the structural frequency vectors $\vec{x}_1,\ldots,\vec{x}_m$. The Fmax *frequency variation* is reflected in the diverse values in $\mathbf{y}$. For the two frequency variations to be highly correlated, typically this means that both reflect (mostly) the same underlying variability caused by some common sources. For example, Figure 3 shows that the two frequency variations reflects the same variability caused by process related sources.

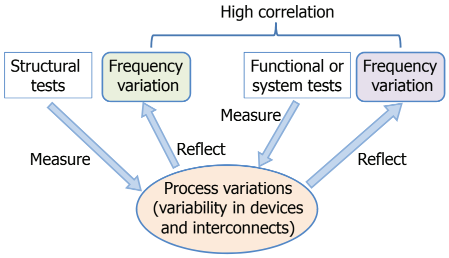

Figure 3. High correlation exists between two freq. variations by reflecting the same underlying variability

On a particular chip sample, the structural delay measurements may not capture exactly what is reflected in the functional or system Fmax. For example, the authors in [6] reported debugging results on some of the speed-limiting paths. In their study, a speed-limiting path is a path whose delay limits the system Fmax on a chip. The authors showed that di/dt voltage droop contributed to the cause of several speed-limiting paths being analyzed. For a chip whose speed-limiting path is contributed by such a cause, it is questionable how much Fmax information can be contained in the results from some structural delay measurements.

From the data learning perspective, information presented on such a chip can be noisy because the sample does not follow the general trend observed across other chips. For building a Fmax predictor, noisy samples can mislead the the data learning process. Therefore, they should be identified and removed from the dataset before the learning.

Figure 4 illustrates an example with two noisy samples. The structural delay measurements are based on two measurements M_1 and M_2. In this example, the n sample chips are grouped into three sets, samples in the D set (dense region), samples in the S set (sparse region), and the two ”noisy samples.” Notice that the two noisy samples are very close in the $\mathbf{X}$ space but the corresponding Fmax values in the $\mathbf{y}$ space are very far. This means that the delay measurement results on these two samples are very similar, while the difference between their Fmax values is much larger than it should be as compared to other samples in the D and S sets, i.e. does not follow the trend.

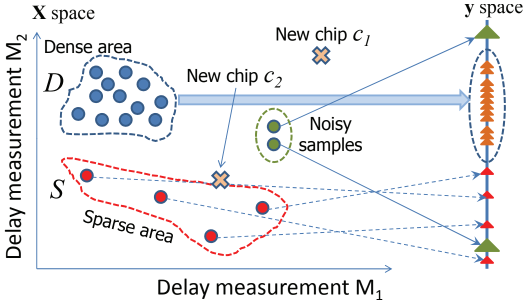

Figure 4. Example to illustrate noisy samples and Fmax predictability from a data learning perspective

Note that a noisy sample needs to be defined with respective to one or more other samples. There needs to be a ”trend” defined first before one can say that a sample does not follow the trend and therefore it is noisy. By definition, we see that a noisy sample is also a hard-to-predict sample by using the trend information presented from other samples. However, a hard-to-predict sample may not necessarily be noisy. A sample can be hard-to-predict because there lacks of the required information in the dataset from which the prediction function P is derived.

For example, suppose P is learned based on set D and set S together and we want to predict the Fmax of a chip c_1. We see that c_1 is very far from any sample used to build P. Because P does not contain much information regarding the region where the new chip c_1 falls into, c_1 can be hard-to-predict by P. A good methodology should include a mechanism to recognize such a situation and raise a flag to the user that c_1 may not be predictable by P. One way to provide such a mechanism is to build a *one-class* learning model [6] that captures the boundary of a ”predictable” region in the $\mathbf{X}$ space. For example, in Figure 4, such a region may be captured as the two irregular areas shown as the dense area and the sparse area. Then, because c_1 falls outside the predictable region, it would be classified as an unpredictable sample.

Suppose we want to use P to predict the Fmax of another chip c_2 that falls on the edge of the sparse area. The one-class model classifies it as a predictable sample because it is inside the predictable region. However, intuitively we can see that the confidence of the Fmax prediction can be low because P may not have enough information in the area close to the sample c_2. To take care of such a situation, a good methodology should also include a mechanism to estimate the confidence of a Fmax prediction.

Figure 5 summarizes the proposed methodology with the two required mechanisms discussed above. As shown in the figure, the Conformity Check selects k conformal chip samples by identifying and removing $m-k$ noisy samples. The k conformal samples are then used to develop a prediction function $P()$ and a conformal model $CM()$ that captures the *boundary* of the predictable region in the $\mathbf{X}$ space. On a

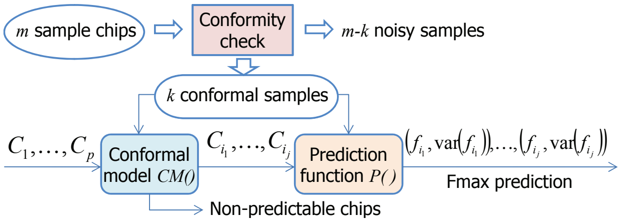

Figure 5. The proposed Fmax prediction flow

given new chip C_i, the conformal model is first applied to check if C_i is predictable or not. If yes, then its Fmax is predicted with $P()$. The prediction result is a pair $(f, var(f))$ where f is the mean Fmax predicted by $P()$ and $var(f)$ is the variance of the predicted Fmax. If C_i does not pass the $CM()$, it is classified as a non-predictable chip.

3 The First Set of Data

The first set of data was collected based on 15 quad-core microprocessor parts by treating each core individually. This resulted in 60 samples, i.e. $m = 60$ in Figure 2. The maximum functional frequencies (Fmax) were measured on these 60 samples. Figure 6 shows the histogram of the Fmax distribution. Frequency variation can clearly be observed.

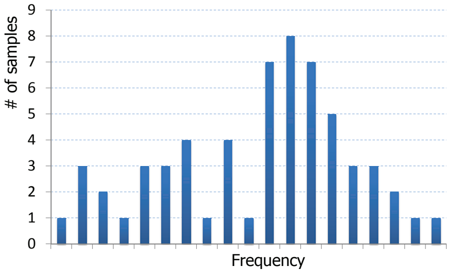

Figure 6. Fmax histogram for the 60 samples

Two test sets were used for the structural delay measurement, a path delay pattern set and a transition fault pattern set. The path delay pattern set was produced based on 12891 testable paths from a total of 24369 timing critical paths reported by the static timing analyzer. One pattern was produced for each testable path using a commercial ATPG tool. The transition fault pattern set was also produced by the ATPG tool (no timing-aware nor n-detection). Based on these two test sets, we consider three types of structural delay measurement as described below. In each measurement, a frequency stepping from low to high was applied and the first failing frequency was recorded.

Flip-Flop (FF) based Using the transition fault pattern set, first failing frequencies were collected on selected 811 scan flip-flops. These 811 FFs were selected because their first failing frequencies could be found through the frequency stepping on all 60 samples. For other FFs, the first failing frequencies might not exist on one or more of the 60 samples. On those FFs even with the highest frequency applied in the frequency stepping, still no failing was observed on some samples.

On every sample, each of the 881 FFs was recorded with a frequency. For each FF j, these recorded frequencies can be represented as a vector $M_j = (x_{1,j}, \ldots, x_{60,j})$ (see Figure 2). The Fmax values is a vector $\mathbf{y} = (y_1, \ldots, y_{60})$. Given the two vectors M_j and $\mathbf{y}$, one can calculate the Pearson correlation coefficient between them $\gamma(M_j, \mathbf{y}) = \gamma_j$. The correlation coefficient γ_j tells how well the frequency variation on FF j correlates to the Fmax frequency variation. Based on these correlation coefficients, we selected the top 100 most-correlated FFs for the experiments. In Figure 2, using 100 FFs means $n = 100$.

Pattern based The first failing frequencies were collected on selected 1331 transition test patterns. The selection of the 1331 patterns is based on the same consideration to the selection of the 811 FFs above. Furthermore, the 100 most-correlated patterns are selected for the experiments. In this case, we also have $n = 100$.

Path based Among the 12891 testable paths, 112 paths were selected and their first failing frequencies were used. Again, the reason for the selection is similar to that discussed above. This gives $n = 112$.

Given data on the three types of structural delay measurement, we were interested in finding out which one could provide the best correlation to the functional Fmax.

4 Initial Correlation Results

In the initial study, we did not consider removing noisy samples. The three types of structural delay measurement were compared based on all 60 samples. To devise an experimental framework to compare them, we took the following three steps:

1. First, we developed an evaluation scheme to assess the quality of a prediction function (or model).
2. Based on the evaluation scheme and the FF-based dataset, we compared the performance of various learning methods for building the prediction function and concluded the best learning method to be used.
3. Based on the best learning method, we then compared the three datasets, the FF-based, the pattern-based and the path-based datasets.

4.1 The evaluation scheme

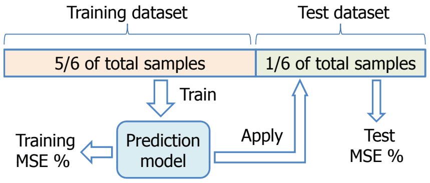

Figure 7. Training and test samples

To evaluation how accurate a prediction function is based on a given dataset, we employ a methodology where the dataset is divided into two subsets, a *training dataset* and a *test dataset* [3]. The training dataset is used to build the prediction function. The prediction accuracy is then evaluated based on only the samples in the test dataset. Using a test dataset that is different from the training dataset tries to avoid the *model over-fitting* problem commonly encountered in data learning [3]. In learning, a over-fitting model means that the model performs well on the samples in the training dataset used to build the model, but performs much worse on the samples in the test dataset.

To avoid statistical bias in the selection of the training and test datasets, in each experiment we repeated the process 50 times with random selection of training and test samples. In each run, the prediction accuracy was evaluated by calculating the Mean Square Error (MSE) on the samples in the test dataset.

Given $(\mathbf{X},\mathbf{y}) = (\vec{x}_1, y_1), \ldots, \vec{x}_{60}, y_{60}))$, the partition of the training and test datasets is illustrated in Figure 7. In each run, the training dataset consisted of randomly selected 50 ($= \frac{5}{6} * 60$) samples, leaving the remaining 10 samples in the test dataset. For simplicity, we use $(\mathbf{X}_r, \mathbf{y}_r) = (\vec{x}_1, y_1), \ldots, (\vec{x}_{10}, y_{10}))$ to denote the test dataset.

Suppose a prediction model f was established based on the 50 training samples. For each test sample $\vec{x}_i$, $i = 1, \ldots, 10$, $f(\vec{x}_i)$ gave the predicted Fmax y'_i. To assess the effectiveness of f on the test dataset, the MSE% was calculated as:

$$\text{Test MSE } \% = \frac{\sqrt{(\sum_{i=1}^{10}(y'_i - y_i)^2)/10}}{|y_{max} - y_{min}|} \tag{1}$$

Notice that the term $(\sum_{i=1}^{10}(y'_i - y_i)^2)/10$ is the MSE on the prediction. The quantity $|y_{max} - y_{min}|$ is the amount of frequency variation across the 60 samples, i.e. the spread along the x-axis in Figure 6. Hence, MSE% estimates the accuracy of f with respect to the actual Fmax variation. Similarly, a Training MSE% can be calculated on the training dataset using the Training MSE $\frac{1}{50}\sum_{i=11}^{60}(f(\vec{x}_i) - y_i)^2)$.

4.2 Selecting the best learning method

We used the evaluation scheme to compare five learning methods: the *k*-NN method and the least-square fit (LSF) method discussed in Section 1.1 before, the ridge regression (RG) method [3], the Support Vector regression (SVR) method [4], and the Gaussian Process (GP) method [5].

For simplicity of the discussion, we omit detailed discussion on each of the learning method. The following gives a high-level summary on their key differences.

Figure 8 summarizes their key differences. The LSF method and the *k*-NN method can be seen as the two basic methods where other methods are derived from. For example, the ridge regression method improves the LSF method by introducing a way to avoid the over-fitting. The RG method not only tries to find a prediction function that best fits the training dataset but also tries to find such a function whose *complexity* is the lowest, i.e. the "simplest" function to best-fit the dataset.

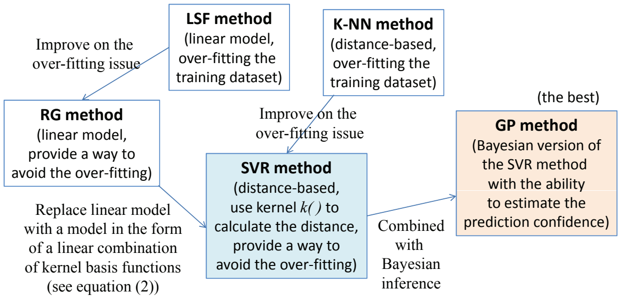

Figure 8. **A high-level summary on the key differences among the five learning methods under study**

While the LSF method is based on a linear equation, the *k*-NN method is based on no equation. In essence, the *k*-NN method is like a table look-up method where the table consists of all training samples and the decision of the look-up is distance based as discussed before in Section 1.1.

The SVR method combines and extends the distance-based idea in the *k*-NN method and the complexity control idea in the RG method. A SVR model always takes the following form [4]:

$$f(\vec{x}) = (\sum_{i=1}^{m'} \alpha_i k(\vec{x}_i, \vec{x})) + a \tag{2}$$

where m' is the number of training samples and $\vec{x}_i$ is the structural delay measurement vector on sample i. The coefficients α_i measures the importance of $\vec{x}_i$ to the prediction. A larger $|\alpha_i|$ means that $\vec{x}_i$ is more important. The quantity $k(\vec{x}_i, \vec{x})$ can be seen as a distance measure between the two vectors $\vec{x}_i, \vec{x}$ through the so-called *kernel* function $k(\cdot,\cdot)$ defined in advance. Each $k(\vec{x}_i, \cdot)$ on a given $\vec{x}_i$ is called a *basis function*. From this perspective, the model can also be seen as a linear combination of the m' basis functions.

In a SVR model, some α's are zero and the corresponding samples become *non-support vectors* and hence, they have no effect on the prediction. For samples with non-zero α's, they are *support vectors*. From this perspective, we see that while the *k*-NN method uses all samples in the prediction, a SVR model uses only the support-vector samples. Moreover, support-vector samples are weighted differently with the weights α's in the prediction.

Gaussian Process [5] is a recently-developed *supervised learning* approach that became popular just in the past few years [5]. The GP method merges the ideas in SVR with the use of Bayesian inference. Hence, GP can be thought as a Bayesian version of the SVR [5].

Like the SVR method, the GP method also measures the importance of each $\vec{x}_i$ in building a model for the prediction. In addition, GP improves from SVR in terms of two major aspects. First, the selection of the kernel function does not have be fixed in advance. Given a family of kernel functions, the GP method is able to automatically select the best one that delivers the best prediction function on a given training dataset. Second, the GP method is able to automatically weight the importance across the n delay measurements $M_1, \ldots, M_n$ and focus on using those more meaningful measurements in the prediction.

The most important property of GP is that unlike the other four methods discussed above, the prediction of a GP predictor is not a single value but a random variable P^* characterized by a Normal distribution $N(f^*(), \sigma()^2)$, where $f^*()$ is the mean of the prediction and $\sigma()^2$ is the variance of the prediction. Hence, with $\sigma()^2$, one can calculate a confidence level on each Fmax prediction.

4.2.1 Comparing the five learning methods

As explained before, we repeated each experiment 50 times with 50 training samples and 10 test samples randomly selected. For each method, we sorted the 50 training sample MSE% results on the 50 runs and obtained an MSE% curve shown in Figure 9. In the figure, five training MSE% curves are shown for the five learning methods. The dataset used in these experiments was FF based.

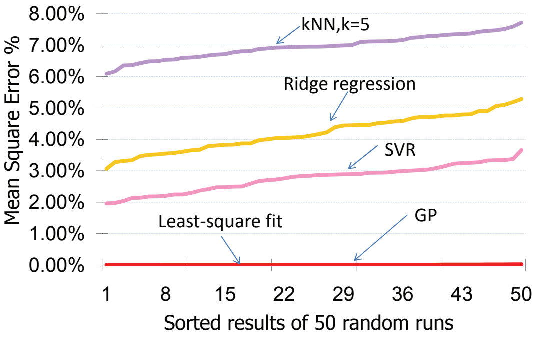

Figure 9. Training sample MSE % - 50 runs with 50 training samples and 10 test samples randomly selected

It is interesting to observe that both least-square fit and GP find almost perfect models to fit the training samples, i.e. the MSE%≈ 0 across the 50 runs. The 5NN method produces the worst result on the training samples.

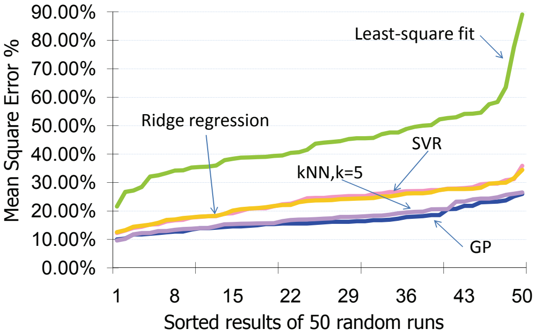

Figure 10. Test sample MSE % - 50 runs with 50 training samples and 10 test samples randomly selected

As discussed before, to compare the effectiveness of the five methods, we should use the MSE% based on the test samples only. The comparison result is shown in Figure 10.

It is interesting to observe that the least-square fit becomes the worst in Figure 10. Furthermore, the prediction accuracy of the least-square fit models vary significantly in the 50 runs. In Figure 10, we also observe that ridge regression and SVR regression performs almost the same. This indicates that changing from a linear model to a non-linear basis-function model does not help much.

The result of 5NN is totally opposite to the result of the least-square fit. 5NN is the worst in Figure 9 but turns out to be almost the best in Figure 10. The good performance of the simple 5NN method and the no improvement result from ridge regression to SVR in Figure 10, indicate that we do not need to use a complex prediction function for the prediction of Fmax values on these samples. This should be considered as a positive result because in data learning, typically the less complex a prediction model is, the higher confidence the prediction model can be applied [4].

From Figure 9 and Figure 10, we conclude that the GP method is the best among the five. GP achieves almost zero error on the training samples, and consistently gives the best prediction results on the test samples in Figure 10. Hence, only the GP method will be used in the experiments below.

4.3 Comparing the three types of measurement

With the GP method, Figure 11 compares the test sample MSE% results based on the three datasets described in Section 3 before. Note that the ”FF-based” curve in the figure is the same result as shown by the ”GP” curve in Figure 10, except that the scales on the y-axis are slightly different in the two figures. Figure 11 shows that the FF-based measurements give the best prediction results.

It is interesting to note that in the two studies in [1][2], both utilized path delay tests. In the pre-silicon design, timing analysis tool identifies critical paths for optimization of design timing. In post-silicon stage, speed limiting paths are identified and improved for pushing performance [6]. These are widely understood methodologies. Hence, it is natural to think that Fmax is largely determined by the delays on some important paths, and if one could find the right set of paths and measure the delays on them, the result should give us a reliable prediction for the Fmax.

Figure 11 tells the contrary. Using structural path delay tests is the least effective way for predicting the Fmax. Observe that the pattern-based dataset is also inferior. The exact reason behind the result in Figure 11 could be complicated. However, an intuitive explanation can be given by comparing the ”information coverage” of the three datasets.

From an information coverage perspective, we observe that the path-based dataset contains the least information while the FF-based dataset contains the most. Recall that the path-based data were collected by measuring delays on 112 selected paths. If we consider the devices and inter-

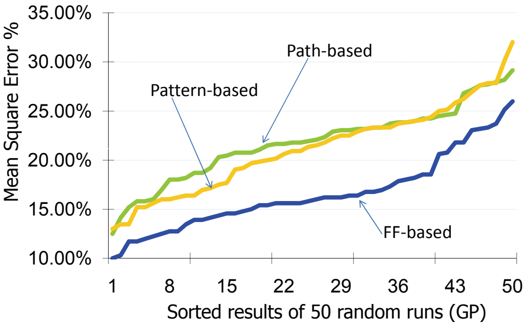

Figure 11. Comparison of three types of dataset

connects covered by these 112 paths, we see that only a very small portion of all devices and interconnects are covered. Unless process variations are largely dominated by systematic with-in die variation, it is unlikely that the characteristics of the devices and interconnects on the 112 paths are representative enough to the characteristics of all devices and interconnects. If Fmax is largely influenced by the characteristics of devices and interconnects not on the 112 paths, the path delay measurements provide not enough information for predicting the Fmax.

Because each transition fault pattern sensitizes many paths, the 100 most-correlated transition fault patterns should cover more devices and interconnects than path delay tests. Hence, more information is presented in the pattern-based dataset.

Each frequency measured on a FF in the FF-based dataset is based on all patterns, i.e. the frequency is the result of checking delays on *all* patterns. Hence, the FF-based dataset provide even more information coverage than the pattern-based dataset. Although more information does not necessarily mean that a better model can be built, at least in the case of Figure 11, this seems to be the case.

5 Removing Noisy Samples

If we take a closer look at the ”FF-based” curve shown in Figure 11, we see that in the 50 runs, the best MSE% result is around 10% and the worst MSE% result is around 26%. The best and the worst differ significantly. This indicates that the Fmax values on some samples are harder to predict by using the information presented from other samples. When these hard-to-predict samples are included in the test dataset, the MSE% result becomes worse.

5.1 The predictability of a sample

For a sample, to estimate how well its Fmax can be predicted by information presented from other samples, we introduce the concept of *predictability*.

We define *predictability measure* as the following. For each sample i with data $(\vec{x}_i, y_i)$, we use the data on the remaining $m-1$ samples, i.e. using the set $S_i = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_{i-1}, y_{i-1}), (\vec{x}_{i+1}, y_{i+1}), \ldots, (\vec{x}_m, y_m)\}$, to build a GP mean prediction model $f*_i()$. Then, we use the quantity $||f*_i(\vec{x}_i) - y_i||$ to tell how predictable the sample i is.

Figure 12 shows the predictability measure results calculated on the 60 samples. Let $r = |y_{max} - y_{min}|$ be the amount of frequency variation across the 60 samples. For convenience, the figure shows a ”15% band” based on $[y_i - 0.15r, y_i + 0.15r]$ with each y_i. One may defines such a band such that for each sample i, if $f*_i(\vec{x}_i)$ is out of the band, it is treated as an hard-to-predict sample.

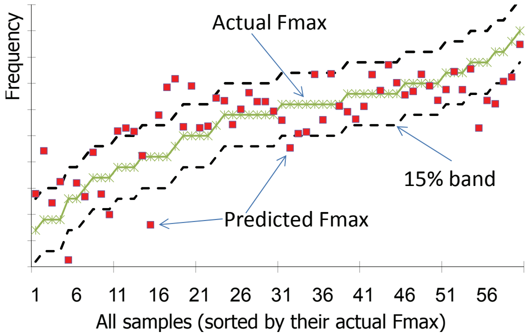

Figure 12. Predictability measure for each sample

Based on the discussion in Section 2 using the example shown in Figure 4 before, a hard-to-predict sample i in Figure 12 can be due to two reasons: (1) There lacks sufficient information in the dataset S_i to predict the Fmax of the sample. (2) Sample i does not follow the trend obtained from the samples in S_i. In the first case, including sample i in learning a predictor may not hurt its prediction accuracy. In the second case, sample i is a noisy sample and including it in learning a predictor will hurt its prediction accuracy. Hence, we desire to remove all samples that fall into the second category.

In order to identify and remove all noisy samples, one straightforward method can be to remove all out-of-the-band samples in Figure 12. However, this method can easily remove many samples that are not noisy. When the initial number of samples is small to begin with, such a method would not be preferred.

Intuitively, if a sample does not follow the trend from other samples, we should see a large prediction error in the predictability measure. Hence, a more conservative strategy is to remove only the hardest-to-predict sample in Figure 12. After this sample is removed, the predictability measure should be re-calculated based on the remaining samples. This is because after removing the hardest-to-predict sample, the predictability measure on the remaining samples can alter significantly. For example, suppose sample 1 is the hardest-to-predict sample. In the original measure, the predictability of sample 2 is based on $S_2 = \{(\vec{x}_1, y_1), (\vec{x}_3, y_3), \ldots, (\vec{x}_m, y_m)\}$. After sample 1 is removed, the new measure is based on $S_2 - \{(\vec{x}_1, y_1)\} = (\vec{x}_3, y_3), \ldots, (\vec{x}_m, y_m)\}$. With the original measure, sample 2 may be hard-to-predict because $\{(\vec{x}_1, y_1)\}$ provides misleading information to predict its Fmax. Hence, by excluding $\{(\vec{x}_1, y_1)\}$ in the new measure, sample 2 may become easy-to-predict.

5.2 Conformity check

Conformity check implements the strategy discussed above in an iterative procedure. In each iteration, the hardest-to-predict sample is identified and removed from the current sample set S_k. A conformal score is calculated for the set S_k with k samples. This conformal score is used to decide when to stop the iteration.

Given k samples, for each sample i, we use the remaining $k-1$ samples to build a GP mean prediction model $f*_i()$. We find the sample j such that $||f*_j(\vec{x}_j) - y_j||$ is the largest across all k samples. Then, this sample j is removed from S_k to form the subset S_{k-1} for the next iteration.

In addition, we calculate the conformal score $CFMC(S_k) = \frac{1}{k}\sum_{i=1}^{k}(f*_i(\vec{x}_i) - y_i)^2$. Beginning with the set of 60 samples S_{60}, Figure 13 shows the result of conformal score calculated at each of the iterations 1-19.

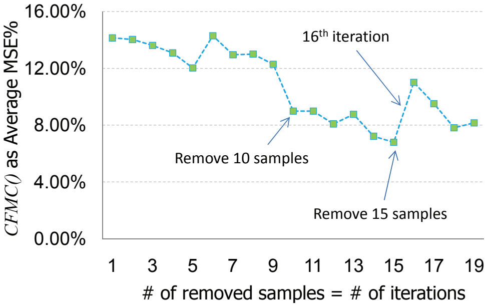

Figure 13. Conformity check and result

For each subset S_{60-j} obtained at the end of the jth iteration, we report the average MSE% as $CFMC(S_{60-j})/|y_{max} - y_{min}|$ where y_{max} and y_{min} are the maximum and minimum Fmax values from the samples in the subset S_{60-j}. Based on the plot, we see that a minimal $CFMC()$ score first happens at iteration 15. Hence, we select S_{45} as a conformal subset. For comparison, we also select S_{50} as another potential conformal subset.

It is interesting to observe that the removal of the 16th sample in the 16th iteration causes a jump in the $CFMC()$ score. This means that the removed sample is actually quite useful for predicting other samples in S_{44}, even though it is the hardest to be predicted using information from other samples in S_{44}. This is a good indication that this sample is actually not a noisy sample.

We compare S_{45}, S_{50}, and the original sample set S_{60}, using the same experimental methodology shown in Figure 7. Again, 50 repeated runs were conducted and we compare them based on their Test sample MSE% consisting of $\frac{1}{6}$ of the total samples in use. Figure 14 shows the result.

Note that the "60 original samples" curve in Figure 14 is the same as the "FF-based" curve in Figure 11, except that the y-axis' scales are slightly different. From Figure 14, we see that the subset S_{45} is better than S_{50}, and both are much better than using the original set. This clearly demonstrates that the conformity check procedure proposed above is doing what it is supposed to do, i.e. producing a subset of samples that a more reliable Fmax predictor can be built on.

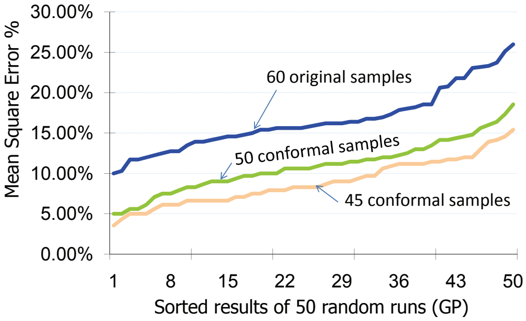

Figure 14. Conformity check produces lower MSE% results with 10 and 15 hard-to-predict samples removed

5.3 Confidence prediction and conformity

As mentioned before, the prediction of a GP predictor for $\vec{x}$ is a random variable with a Normal distribution $N(f^*(\vec{x}), (\sigma(\vec{x}))^2)$. Because the prediction is a Gaussian random variable, its confidence range can be calculated with $\sigma(\vec{x})$ and a Normal distribution table. For example, $f^*(\vec{x}) \pm 3\sigma(\vec{x})$ corresponds to 99.7% confidence range and $f^*(\vec{x}) \pm 2\sigma(\vec{x})$ corresponds to 95.4% confidence range.

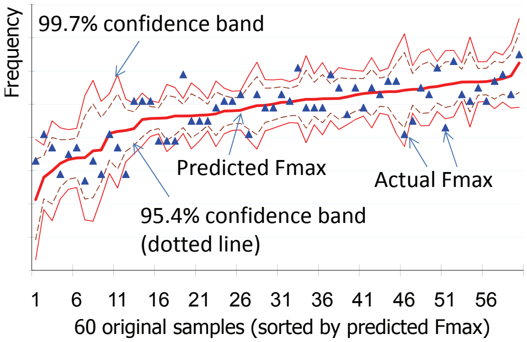

Figure 15. Confidence prediction by Gaussian Process on the 60 original samples, the 99.7% band is much looser than that shown in Figure 16 below

Figure 15 shows the result similar to the predictability measure result in Figure 12 based on the 60 original samples, i.e. each sample i is predicted based on the GP predictor $f*_i()$ built from the remaining 59 samples. In Figure 15 the "99.7% confidence band" and "95.4% confidence band" replace the "15% band" in Figure 12. Observe in Figure 15 that most of the actual Fmax points stay inside the 99.7% band, in contrast to in Figure 12 where many predicted Fmax points stay outside the constant ±15% band. Notice the difference that in Figure 15 the x-axis is based on sorting the predicted Fmax values while in Figure 12 the x-axis is based on sorting the actual Fmax values.

Figure 16 shows the result similar to Figure 15, based on the 45 conformal samples. Notice that not only the prediction accuracy improves but also the 99.7% band becomes a lot tighter (the scales on the y-axis in the two figures are

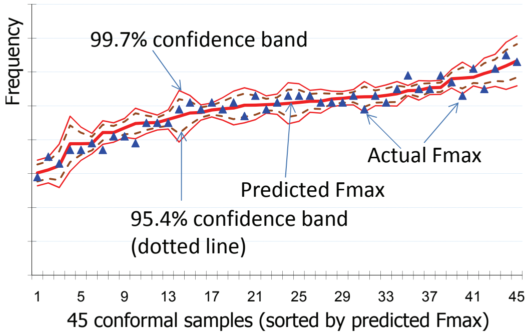

Figure 16. **Confidence prediction by Gaussian Process on the 45 conformal samples, the 99.7% band is much tighter than that shown in Figure 15**

the same). This is a very interesting result showing that using the conformity check to select samples not only can improve the accuracy of the resulting predictor but also can improve the confidence of the prediction.

By comparing the confidence results in Figure 15 and in Figure 16, we observe that the confidence estimations by the GP predictors make sense. In Figure 15 where prediction errors tend to be larger, the confidence bands are also larger indicating less confidence on the prediction. In Figure 16 where prediction errors tend to be smaller, the confidence bands are also smaller indicating more confidence. Therefore, we see that the GP predictors capture very well the trend of the prediction error in its confidence estimation.

6 Targeting on System Fmax

The second set of data was collected based on 79 new samples. The system Fmax was measured on each sample. The structural delay measurement data was based on 1507 FFs similar to the FF-based dataset described before.

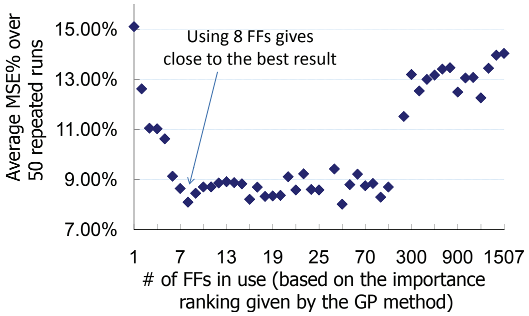

Figure 17. **Selecting the most-correlated FFs shows that using only 8 FFs gives the best result**

Instead of using data on all 1507 FFs, we selected the 8 most-correlated FFs based on applying the GP method first to the complete dataset consisting of all 1507 FFs and the 79 samples. As mentioned before, in building a GP predictor, the GP method would also evaluate the importance of each FF measurement based on how important the frequency variation on the FF is for predicting the Fmax. With this importance value, we can rank all 1507 FFs and select the top i most important FFs to use. Figure 17 show the average MSE% result by using the top i FFs selected by GP, for $i = 1, 2, \ldots, 1507$. Again, this MSE% is based on the evaluation scheme discussed in Section 4.1 before. Observe that using only 8 FFs can deliver almost the best result.

Using the 8-FFs dataset, Figure 18 shows the conformity check result similar to that shown in Figure 13 before. Notice that the average MSE% results in this figure are smaller than those shown in Figure 13. This indicates that in general, Fmax values on these 79 new samples are more predictable than the 60 samples studied before. Moreover, the curve is smoother than that in Figure 13, indicating that most of the removed samples are indeed noisy samples.

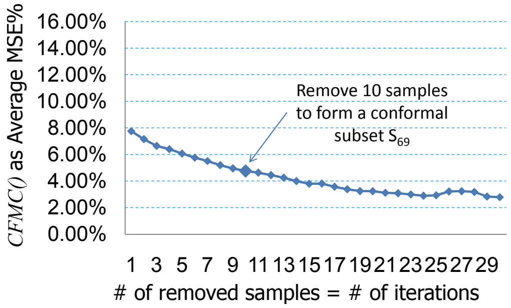

Figure 18. **Conformity check result w/ 8-FFs dataset**

For illustration, we arbitrarily select the removal of 10 samples. Figures 19 and 20 then show similar results to those in Figures 15 and 16 before, where once again, removing samples by the conformity check improves the Fmax predictability as well as the confidence of the prediction.

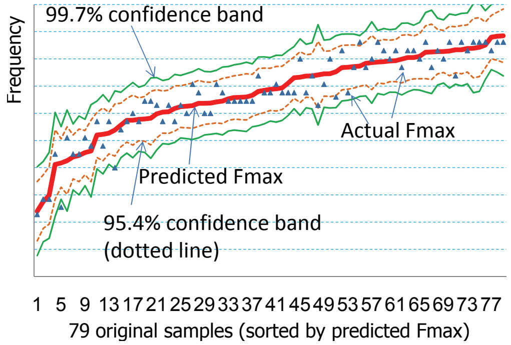

Figure 19. **Result on the 79 original samples**

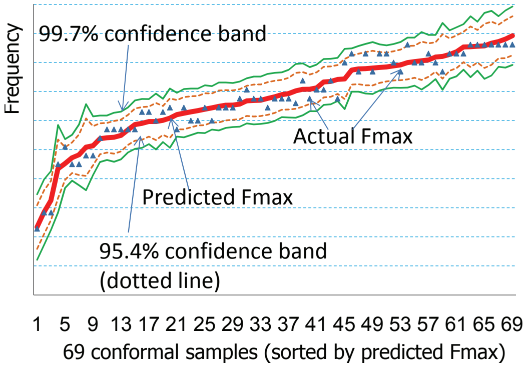

Figure 20. **Result on 69 conformal samples**

7 Building a Conformal Model

Given a set F of conformal samples, the goal of a conformal model CM is to capture the characteristics of the conformal samples so that on any given new sample $\vec{x}$, we can decide if it is also a conformal sample or not. In a sense, CM captures the "boundary" of the set F in the $\mathbf{X}$ input space. Learning a model for a single class of samples is a 1-class unsupervised learning problem and we can use the 1-class SVM algorithm to solve the problem. The 1-class SVM algorithm was studied extensively in [7] before. Based on the 45 conformal samples from the first set of data, we applied 1-class SVM to build a model $CM_{45}()$. For each sample $\vec{x}_i$, $CM_{45}(\vec{x}_i)$ returns a score telling where the sample falls with respective to the conformal set boundary.

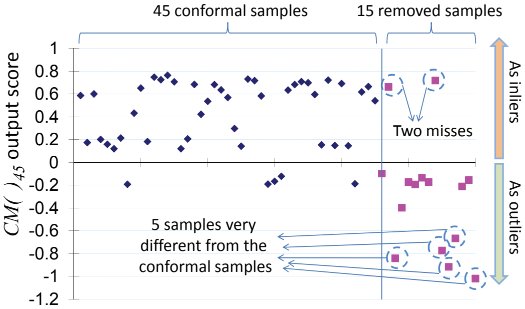

Figure 21. **Using $CM()_{45}$ to screen samples**

Figure 21 shows the scores for the 60 samples. Samples with scores above 0 are inside the boundary. We see that the model identifies 13 of the 15 samples removed by the conformity check as outside the boundary, while miss-classifying 5 conformal samples as being outside and 2 removed samples as being inside. Also notice that 5 samples have scores further down below the 0 threshold, which show that their characteristics are very different from the conformal samples. In application, the conformal model is used to screen out samples as shown in Figure 5. Hence, the 5 miss-classified conformal samples can be thought as "over-kills" and the two missing samples as "escapes." The problem on the two escaping samples is more severe because their Fmax values are likely to be miss-predicted.

We note that in our methodology any escaping sample can still be identified as a low-confidence sample in the GP prediction. For example, in Figure 22 we show that in 50 repeated runs, the average variance $\sigma()^2$ output from the GP predictors on the two escaping samples tends to be much larger (less confidence) than the average variance on the 7 test samples randomly selected from the conformal set.

Figure 21 and Figure 22 together show that in application of a GP predictor built based on a conformal set of samples, there are two steps to screen out samples that are likely to be miss-predicted: (1) A conformal model can identify those samples as outside the conformal boundary. (2) In the actual GP prediction, a sample with very low confidence can be discarded. These two methods minimize the chance that a conformal-set GP predictor is misused in prediction for samples not conformal to the training set and consequently, improve the robustness of the Fmax prediction.

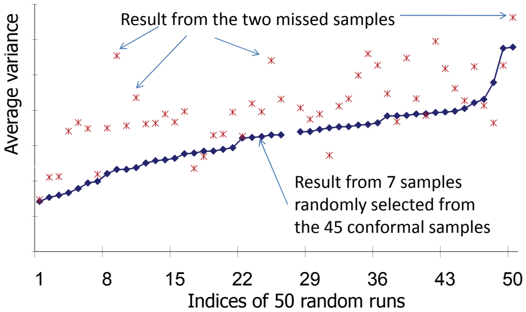

Figure 22. **The two missed samples tend to have low confidence (large variance) in GP prediction**

8 Conclusion

In this work, we propose a data learning methodology for predicting functional or system Fmax based on structural delay test measurements. We discuss various data learning techniques to implement this methodology and present experimental results to explain these techniques based on datasets collected from a recent microprocessor design. We suggest a new method called *conformity check* that can be used to prune a dataset so that a more reliable Fmax predictor can be built on. We studied various learning methods and determined that the Gaussian Process method is the most effective one. We present results based on two sets of data, one targeting on functional Fmax and the other targeting on system Fmax. We show similar results given by the conformity check on both sets of data and confirm that the proposed methodology can be applied in both situations. Finally, we discuss how one-class learning can be used to build a conformal model for screening samples. The use of such a conformal model and the confidence prediction provided by a GP predictor together ensure the robustness of the proposed Fmax prediction methodology.

References

[1] Cory, B.D.; Kapur, R.; Underwood, B. Speed binning with path delay test in 150-nm technology. IEEE Design & Test of Computers, Volume 20, Issue 5, pp 41 - 45.

[2] Jing Zeng, et al. On correlating structural tests with functional tests for speed binning of high performance design. *ITC*, 2004, pp. 31-37.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning - Date Mining, Inference, and Prediction. *Springer Series in Statistics*, 2001

[4] Bernhard Schlkopf, and Alexander J. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, 2001.

[5] Carl E. Rasmussen and Christopher K. I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2005.

[6] P. Bastani, et al. Speedpath prediction based on learning from a small set of examples. *Proc. DAC*, 2008, pp. 217-222.

[7] Sean H. Wu, D. Drmanac, Li-C. Wang. A Study of Outlier Analysis Techniques for Delay Testing. *ITC*, 2008.