Simplified Subspaced Regression Network for Identification of Defect Patterns in Semiconductor Wafer Maps

Fatima Adly, Omar Alhussein, Paul D. Yoo, *Senior Member, IEEE*, Yousof Al-Hammadi, Kamal Taha, *Senior Member, IEEE*, Sami Muhaidat, *Senior Member, IEEE*, Young-Seon Jeong, Uihyoung Lee, and Mohammed Ismail, *Fellow, IEEE*

***Abstract*—Wafer defects, which are primarily defective chips on a wafer, are of the key challenges facing the semiconductor manufacturing companies, as they could increase the yield losses to hundreds of millions of dollars. Fortunately, these wafer defects leave unique patterns due to their spatial dependence across wafer maps. It is thus possible to identify and predict them in order to find the point of failure in the manufacturing process accurately. This paper introduces a novel simplified subspaced regression framework for the accurate and efficient identification of defect patterns in semiconductor wafer maps. It can achieve a test error comparable to or better than the state-of-the-art machine-learning (ML)-based methods, while maintaining a low computational cost when dealing with large-scale wafer data. The effectiveness and utility of the proposed approach has been demonstrated by our experiments on real wafer defect datasets, achieving detection accuracy of 99.884% and R^2 of 99.905%, which are far better than those of any existing methods reported in the literature.**

***Index Terms*—Ensemble learning, machine learning (ML), semiconductor wafer defect detection, semiparametric models.**

I. Introduction

SEMICONDUCTOR industry has been witnessing an astounding growth rates over the past few decades. The industry sales in the past two years alone have been estimated to exceed 300 billion USD worldwide, as reported by Semiconductor Industry Association (SIA) [1]. The growth rates in this industry are moving in an escalating manner, but with such advances in the manufacturing process, comes complicated designs, requiring a highly complex fabrication process consisting of hundreds of steps [2]–[4]. During any of the steps, a defect might occur, causing a chip malfunctioning. These defects can be as a result of imperfections in the machines used, or due to the chemical dyes used, physical damages, human mistakes, etc. Such defects can be visualized using wafer maps, as they leave unique patterns on the chip [5]. Identifying these patterns improves the chances of fixing the point of manufacturing failure, thereby reducing the company's losses [3].

Several machine-learning (ML) techniques have been used in the semiconductor field; however, the widely used models are artificial neural networks (ANNs) and support vector machines (SVMs). They have recently gained much attention because of their powerful model-free nonparametric learning ability. Chen and Liu [6] implemented an ANN technique called artificial resonance theory (ART1) on actual data obtained from a semiconductor manufacturing company in Taiwan. They compared their work with a different ANN technique, an unsupervised self-organizing map (SOM). They showed that the training time required by ART1 was 10 times faster than that of SOM, and that ART1 had better classification abilities in detecting similar defect patterns. Choi *et al.* introduced a further improvement to the ART1 algorithm by proposing a MultiStep-ART1 that includes similarity evaluation and data preprocessing scheme [7]. Experiments showed that the proposed method was capable of identifying mixed patterns on wafers, and was able to classify patterns with higher accuracy and lower computational time as compared to ART1. Chang *et al.* [8] implemented a contextual-Hopfield ANN and inspected the dies in multiple stages. They were able to achieve 92.4% accuracy by taking results from every step and recording them in a die map.

As for SVM, the goal behind it is to design a hyperplane that classifies all training vectors into different classes. Many planes can be suitable for the same problem, in such cases the one that leaves maximum margin between the classes is chosen [9]. SVMs are widely and frequently used by many researchers, as they can be easily manipulated. SVM has been combined with regression models as implemented by Xie *et al.* [10], and it has also been shown to be useful for reliability analysis, as demonstrated by Song *et al.* [11]. Baly and Hajj [12] evaluated the performance of different ML models including SVM and general regression neural network (GRNN). The results of their

Manuscript received October 22, 2014; revised May 07, 2015; accepted September 01, 2015. Date of publication September 23, 2015; date of current version December 02, 2015. This work was supported by Information and Communications Technology (ICT) Fund. Paper no. TII-14-1166.

F. Adly, Y. Al-Hammadi, K. Taha, S. Muhaidat, and M. Ismail are with the Advanced Technology Investment Company (ATIC)-Khalifa Semiconductor Research Center, Khalifa University, Abu Dhabi, United Arab Emirates, and also with the ECE Department, Ohio State University, Columbus, OH USA (e-mail: fatima.alshawish@kustar.ac.ae; yousof.alhammadi@kustar.ac.ae; kamal.taha@kustar.ac.ae; sami.muhaidat@kustar.ac.ae; ismail@kustar.ac.ae).

O. Alhussein is with the School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (e-mail: oalhusse@sfu.ca).

P. D. Yoo is with the Department of Computing and Informatics, Bournemouth University, Poole BH12 5BB, U.K. (e-mail: paul.d.yoo@ieee.org).

Y. S. Jeong is with Chonnam National University, Gwangju, South Korea (e-mail: young.jeong@chonnam.ac.kr).

U. Lee is with the Memory Division, Samsung Electronics Co., Hwasung, South Korea (e-mail: uihyoung.lee@samsung.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2015.2481719

experiments illustrated that SVM model outperforms all other models by means of classification accuracy, and it achieved the lowest false negative prediction rates. Xie *et al.* [13] proposed a MultiClass-SVM. Based on their experimental results, they claimed that MultiClass-SVM has superior performance to ART1 and SOM in terms of identifying defect patterns with different sizes, locations, and angles; however, they are similar in shape. They also found that MultiClass-SVM was more robust in the presence of noise.

In general, ANNs were used for their simplicity, good ability to handle nonlinear and multidimensional problems. They are considered to be both fast and adaptive learners, in the sense that the network changes its structure based on the available information during the training process [14], [15]. SVM models have the ability to deal with multiclass classification problems, multimodal data, and with datasets consisting of inseparable points [12]. Despite the fact that these models are widely used in the literature, they still suffer from various limitations. For example, with ANN, the two main drawbacks would include the high computational complexity, especially when handling large datasets, and susceptibility toward over-fitting problems [16]. SVM even requires longer training time since the number of variables used in SVM equals the number of training data. SVM is optimality, where in an optimal model, the parameters used must be chosen carefully such that it includes the kernel selection and tuning its parameters, margin value, gamma value, etc. Such processes could be time consuming and it might cause model generalization problem [17].

In this paper, we introduce two novel techniques namely, randomized general regression and simplified subspacing in Methods I and II, respectively. In Method I, a novel algorithm that combines a general-regression-based consensus learning model with a powerful randomization technique is designed to achieve an accurate and efficient classification of wafer defect patterns. Method I works as a large collection of decorrelated fast general regression learning models, such that each learner depends on the values of a random vector sampled independently and with the same distribution for all learners in the learning network. This approach leads to models with low variance and high accuracy. In addition, due to randomized learning process, the model generalization ability is significantly improved; as memorizing the data is not an option anymore, and the generalization error for Method I converges as the number of learners becomes large. Method II converts input vector $\boldsymbol{x}$ to the centroid vector $\boldsymbol{c}$, ultimately reducing input data significantly. Method II is implemented through a data partitioning and clustering like algorithm creating a final simplified subspaced regression network (SSRN). *Voronoi*-based data partitioning clusters the original dataset, then the K-means takes the center of each cluster. With SSRN, thus, the time complexity of the model is improved significantly. To validate our work, the classification performance of the proposed methods is compared to those-related ML literatures, using experimental verification based on model accuracy, stability, and time complexity.

This paper is organized as follows. Section II describes the wafer defect data used and the proposed methods. Sections III discusses the testing results obtained from comparing different ML -based models against the proposed methods. Section IV concludes.

II. Data and Method

Our experiments consist of four consecutive phases.

1) Generating and collecting the data of defective patterns appearing on semiconductor wafers.
2) Construct ML models and tuning their parameters. In this phase, SVM, sequential minimal optimization (SMO), ANNs including GRNN, radial basis function (RBF), probabilistic neural network (PNN), and multi-layer perceptron (MLP), the proposed consensus models with Methods I and II are constructed through set of experiments that help to choose the proper parameters.
3) The predictive performance of the proposed methods is compared against those of SVM and ANNs for accuracy (Acc.), coefficient of determination (R^2), variance (σ) for model stability and generalization ability, and time to build the model (TBM) for time complexity.
4) Finally, we compare those results with the consensus results from the literature.

A. Defect Pattern Generation

The commonly appearing wafer defect patterns are considered in this study. The dataset used throughout the experiments conducted in this paper is a mixture of real data provided by Samsung Electronics in Korea, and generated data by Jeong *et al.* [18] based on the approach introduced by DeNicolao *et al.* [19]. Four different patterns of wafer defects were generated, namely, circle, cluster, repetitive, and spot. For every defect pattern, a different probabilistic model was used; the models represent the failure position of a die on the wafer. The probabilistic expressions used are demonstrated by the following equations [19].

Circle: The controlling parameters are the width of the circle σ, and its center (x_c, y_c)

$$p(x,y) = 1 - \exp\left(r^2/2\sigma^2\right), r^2 = (x-x_c)^2 + (y-y_c)^2. \tag{1}$$

Cluster: Generated by performing logical "AND" or "OR" operations, in which the failure can be a combination of two patterns

$$(\text{"AND"}) : p(x,y) = p_1(x,y)\, p_2(x,y) \tag{2}$$

$$(\text{"OR"}) : p(x,y) = p1(x,y) + p2(x,y) - p1(x,y)\, p2(x,y). \tag{3}$$

Repetitive: Generated by repetitive rows/columns, and controlled by the location of the rows/columns, which are T and ϕ

$$(\text{horizontal}) : p(x,y) = (1 + \sin(2\pi y/T + \phi\phi))/2 \tag{4}$$

$$(\text{vertical}) : p(x,y) = (1 + \sin(2\pi x/T + \phi\phi))/2. \tag{5}$$

Spot: The controlling parameter are its width σ, and center (x_c, y_c)

$$p(x,y) = \exp\left(r^2/2\sigma^2\right), r^2 = (x-x_c)^2 + (y-y_c)^2 \tag{6}$$

where x, y are the center coordinates of the wafer, and r is the distance between the center of the wafer and the defect's center.

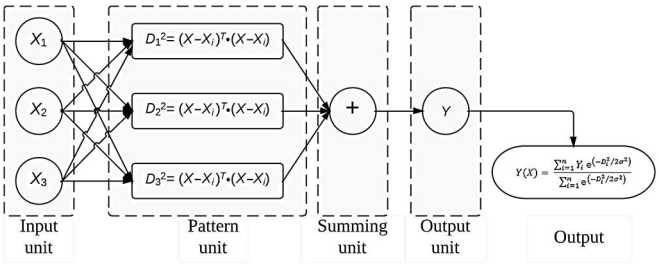

Fig. 1. GRN architecture. It consists of four layers: input, pattern, summation, and output.

B. Method I: Randomized General Regression

In this method, we apply a randomization technique to the general regression network (GRN). The randomization can be seen as bootstrap aggregation, which attempts to improve the classifier performance by combining copies of the original classifier but with randomly selected data subsets with replacement. This is mainly due to the fact that the performance of combined classifiers outperforms any single classifier working independently. Method I randomization technique depends on bootstrap statistical techniques, in which the original data are sampled in order to create n-new random subsets of data with replacement, where n represents the number of classifiers that shall be combined. Hence, each classifier will be trained using a different dataset [20]–[24]. The classifier chosen is GRN, which is kind of a probabilistic neural network that makes decisions based on statistical approach [23], [24]. GRN is made up of four layers: 1) an input layer; 2) a pattern layer; 3) a summation layer; and 4) an output layer, as demonstrated in Fig. 1.

The pattern layer calculates the distance between the input and prediction vectors, giving an indication of how well the prediction values represent the actual ones. The distance is calculated based on the following equation:

$$D_i^2 = (X - X_i)^T \cdot (X - X_i) \tag{7}$$

where X and X_i correspond to the sample to predict and the training sample, respectively. Then, the summation layer takes the output from the pattern layer and performs two operations. 1) Sum up the product of the weights and the target output of every neuron in the pattern layer. 2) Sum of the individual weights. The sum of each operation is then sent to the output layer, where a final decision is made using the following equation:

$$Y(X) = \frac{\sum_{i=1}^{n} Y_i \mathrm{e}^{\left(-D_i^2/2\sigma^2\right)}}{\sum_{i=1}^{n} \mathrm{e}^{\left(-D_i^2/2\sigma^2\right)}} \tag{8}$$

where Y is the estimation of the observed outputs, Y_i multiplied by an exponential term based on the previously calculated Euclidean distance, and averaged over the number of output neurons n, and σ denotes the controlling (smoothing) parameter of how closely the estimation fits the data. Fig. 3 depicts the pseudocode of Method I, followed by Fig. 2, the general architecture.

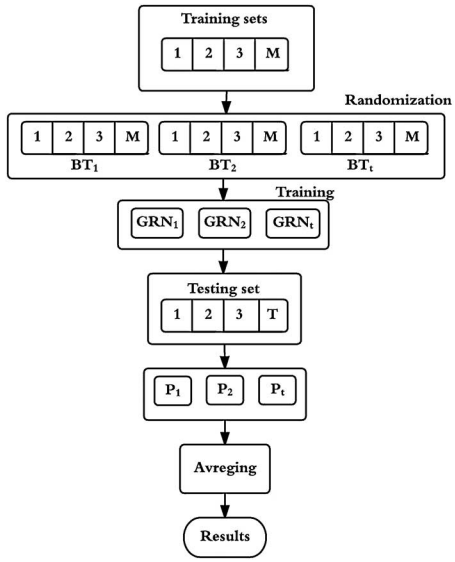

Fig. 2. General architecture of Method I. Randomized bootstrap data are generated, each dataset is being used by different GRN models. Randomly selected subset with replacement is applied to each GRN, and the final classification output is the majority vote taken from all GRN models.

Algorithm 1

Data: D : Training data;
Data: T : Testing data;
Data: K : Number of cross validation folds;
Data: M : Number of weak learners;
Coding the classes;
Partition D into K folds k_1, k_2, k_3… k_K
For i from 1 to K, do
  Initialize training data set for cross validation;
  Use k_i for training and remaining for testing;
  For j from 1 to *M*, do
    Generate bootstrap samples;
    Train GRN network using bootstrap sample;
  end
  For n from 1 to *M*, do
    Predict output of *M*-learners using *T* data;
  end
  prediction = un-weighted majority voting;
  accuracy measurements;
end

Fig. 3. Randomized general regression neural network (RGRN) pseudocode.

C. Method II: Simplified Subspacing

Method II proposes a data reduction technique based on efficient data partitioning and clustering algorithms. The Voronoi diagram (VD) is used as the data partitioning mechanism that clusters the whole vector space into smaller Voronoi regions. K-means algorithm then is used to fetch the centroid of each Voronoi region to be used as the representative of all original vectors. By using only the centroids, the size of the data is significantly reduced; hence, the time complexity of the model shall be improved. In addition, combining with parametric partitioning model introduces semiparametric properties to the pure nonparametric learning model. Fig. 4 illustrates the stepwise procedure of Method II.

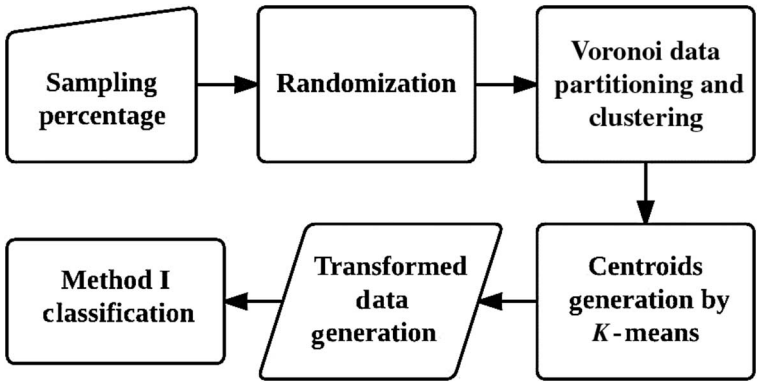

Fig. 4. Stepwise procedure of Method II. SRS is performed based on the specified sampling percentage. Voronoi and K-means algorithms are applied to reduce the data size. Then, the new dataset is used by SSRN for classification.

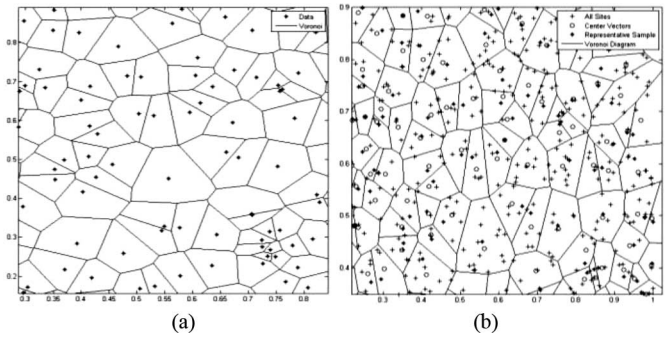

Fig. 5. Data partitioning Voronoi. (a) Cell partition. (b) k-means centroids.

The first stage of the algorithm is to perform simple random sampling (SRS) on the dataset to select n representative samples out of a total of N samples. SRS selects n equiprobable random samples from the original dataset with replacement following a binomial distribution [25]. The second stage is to partition the dataset space into n *Voronoi* cells, based on the selected representative samples. There exist many efficient implementation methodologies for the VD, such as lower bound, divide-and-conquer, Fortune's line sweep, and brute force. VD can be efficiently constructed using sweeping algorithm, since it is a simple algorithm with $O(n \log n)$ time complexity, where n in the number of segments in the plane [26]. In VD, a representative subset $S = \{p_1, p_2, \ldots, p_n\} \in R^{\text{Data Dimension}}$ is used to partition the dataset space into n-*Voronoi* cells with a sampling percentage of $\frac{n}{N}$ as defined by the following equation:

$$Vor\,(p_i) = \{x \| \|xp_i\| \leq \|xp_j\| \; \forall j \in S\} \tag{9}$$

where $\|xp_i\|$ represents the Euclidean distance operator. Fig. 5(a) illustrates the VD applied on some random 2-D dataset.

Upon performing VD on S, we overlay the entire (original) dataset onto the generated Voronoi space. And to decide whether a point belongs to the cell corresponding to the point p_i (Vp_i), where Vp_i represents the *Voronoi* cell of the point p_i, the Euclidean distance is calculated between the new point and every point in S, and the decision is made based on minimum distance, i.e., a point a belongs to $V(p_i)$ iff $d\,(a, p_i) < d\,(a, p_j)$ where $i \neq j$. By adopting this technique, the time complexity of classification shall be significantly reduced [26]–[28]. Then, the centroid of each Voronoi cell is fetched using the K-means algorithm, as shown in Fig. 5(b). Thereby, the data size is reduced to the previously specified sampling percentage. This marks the end of the preprocessing stage, where RGRN is performed on the new reduced dataset to perform classification. The pseudocode of the Method II is illustrated in Fig. 6.

Algorithm II

Input: Dataset S of n-data points
1) sp = Specify the sampling percentage
2) Desired No. of clusters = percentage of data included in sampling
Output: Classification results
***3)* Functions and variables used:**
Sd: Randomly sampled dataset based on sp
4) Vor(Sd): Construct the Voronoi diagram of Sd
5) v(p): Voronoi cell of point p
Neighbor(p): all neighbors of point p

Generate Sd via SRS
6) Call Vor(Sd)
Select random point p from Sd
7) Call Neighbor(p)
Get centroid of each cluster via k-means algorithm
8) Extract the reduced dataset (Rd)
9) Use Rd as input data of RGRN algorithm
10) Get classification results
11) Stop

Fig. 6. RGRN pseudocode.

D. Model Validation and Testing

Model validation is of high importance, as we need to be perfectly sure that our system performs as desired. This becomes more crucial when dealing with dynamic systems that are dependent on observing the data rather than relying on static set of equations to give decisions, which is the case in ML models. Several methods were used to validate the proposed methods using theoretical relationships and experimental analyses. To assess the predictive performance of the underlying models, we perform k-fold cross-validation, which divides the data into k-folds, uses $k-1$ for training and remaining one for testing k

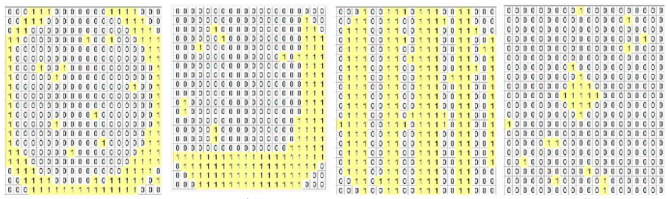

Fig. 7. Defect patterns. (a) Circle. (b) Cluster. (c) Repetitive. (d) Spot.

times [29], [30]. To validate the quality of the formed clusters in the SSRN model, we compute the F-measure by calculating the harmonic mean of the specificity and sensitivity of the data, where its value falls in the range between 0 and 1. Hence, if the F-measure value after the clustering is close to what it was before the clustering, then the selected centroids of each cluster were indeed good representatives of the data [31], [32].

III. Model Evaluation and Analysis

The underlying models are evaluated using the same dataset. The data, as mentioned in Section II, are a mixture of real and generated data, where a 20×20 wafer maps are constructed, i.e., 400 chips per wafer. For each defect pattern, 80 wafer maps are generated by manipulating the controlling parameters of the probabilistic model of each defect pattern, to get variations of the same pattern. For example, the same defect pattern but different locations, sizes, thicknesses, etc. During the generation process, several assumptions were made such as setting the wafer radius to 1, and the wafer center to the origin of the plane. Fig. 7 shows that the defect patterns used circle, cluster, repetitive, and spot, respectively. The wafer maps shown in Fig. 7 were generated using a comparison technique that finds how many defective chips are surrounding a functional chip, and *vice versa*. This was generated based on the joint-count statistics technique as implemented by Jeong *et al.* [18].

The predictive performance of the models considered in this study is measured by model accuracy (Acc.: comparing the predicted output with the actual one), variance (σ: which is a measurement of deviation of every data point in data set from the mean, which gives an indication of model stability and generalization ability), coefficient of determination (R^2: which indicates the model's capability to explain and predict future outcomes correctly), and time complexity (TBM in seconds). The above measures can be obtained using the following equation:

$$\text{Acc.} = \frac{\text{length}\left(X = \hat{X}\right)}{\text{length}\,(X)} \tag{10}$$

where X is the correct classification value and $\hat{X}$ is the estimated one

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}{n-1}} \tag{11}$$

$$R^2 = 100 \times \left(1 - \frac{\sum_{i=1}^{n}\left(x_i - m_i\right)^2}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}\right) \tag{12}$$

TABLE I
Performance of Methods I and II

Model	Fold	Acc.	R^2	σ	TBM
Method I	7	99.702	99.048	1.127	10.755
	8	99.739	99.167	1.127	11.575
	9	99.537	99.213	1.128	14.692
	10	**99.792**	**99.333**	**1.128**	**15.774**
Method II	7	98.671	97.290	1.282	7.583
	8	98.837	99.288	1.248	8.957
	9	99.742	99.789	1.260	9.364
	10	**99.884**	**99.905**	**1.257**	**10.930**
MLP	7	79.063	75.169	5.604	30.910
	8	**80.000**	**74.460**	**6.328**	**27.380**
	9	77.813	78.730	5 .571	31.020
	10	79.375	74.287	6.826	29.210
RBF	7	91.667	93.939	0.833	84.311
	8	91.667	78.181	0.726	92.176
	9	**96.667**	**92.727**	**0.834**	**85.214**
	10	93.750	90.909	0.873	120.122
PNN	**7**	**95.833**	**96.667**	**1.179**	**1.829**
	8	85.000	76.000	1.261	1.890
	9	90.000	80.000	1.050	1.927
	10	93.750	95.000	1.209	1.987
GRN	**7**	**95.833**	**96.667**	**1.179**	**2.966**
	8	85.000	76.000	1.261	1.908
	9	90.000	80.000	1.050	2.073
	10	93.750	95.000	1.209	2.178
SVM	7	90.939	85.461	1.104	0.987
	8	91.250	85.250	1.113	1.434
	9	**91.279**	**83.736**	**1.101**	**1.206**
	10	90.938	83.500	1.106	1.256
SMO	7	91.563	82.883	3.579	0.180
	8	**92.188**	**82.919**	**4.016**	**0.190**
	9	91.250	82.974	3.993	0.180
	10	91.563	82.937	4.289	0.180

The parameters of each model were given the following values: Method I-RGRN (numWeakLearners: 7, numCvFolds: 12, numInputNeurons: 294, spread: 0.1), Method II-SSRN (numWeakLearners: 7, numCvFolds: 10, numInputNeurons: 294, spread: 0.1), MLP (learningRate: 0.09, trainingTime: 30, validationThreshold: 30), RBF (numInputNeurons: 100, spread: 7), PNN (numInputNeurons: 272, spread: 3), GRN (numInputNeurons: 272, spread: 2.75), SVM_{Lib} (numCvFolds: 10, kernelType: polynomial, SVMType: C-SVC, gamma: 0.07, shrinking: false), and SMO (numCvFolds: 10, C-value: 2, kernelType: rbf).

where n is the data size, x_i is the actual value of the data, $\bar{x}$ is the mean, and m_i is the predicted output obtained from the network model. The experimental setup used is a 64-bit operating system, processor speed of 2.6 GHz Intel CPU core i5, and installed memory (RAM) of 8 GB DDRL3 running on the platform of Microsoft Windows 7 Enterprise.

Table I illustrates the performance comparisons of the proposed models (Methods I and II) against those of SVM, SMO, and ANNs including GRN, RBF, PNN, and MLP. The best score in each category is underlined, and the best fold scores in each model are bolded. As seen in Table I, the proposed models outperform on 10 folds, achieving a far better Acc. and σ than the rest of the models. Methods I and II achieved 99.792% and 99.884% model accuracies, respectively. Yet, their computational cost is slightly larger. Fig. 8 visualizes the results obtained in Table I for 10 folds. Based on the obtained experimental results, both proposed methods proved to be superior to all other ML models. Moreover, they outperformed other defect

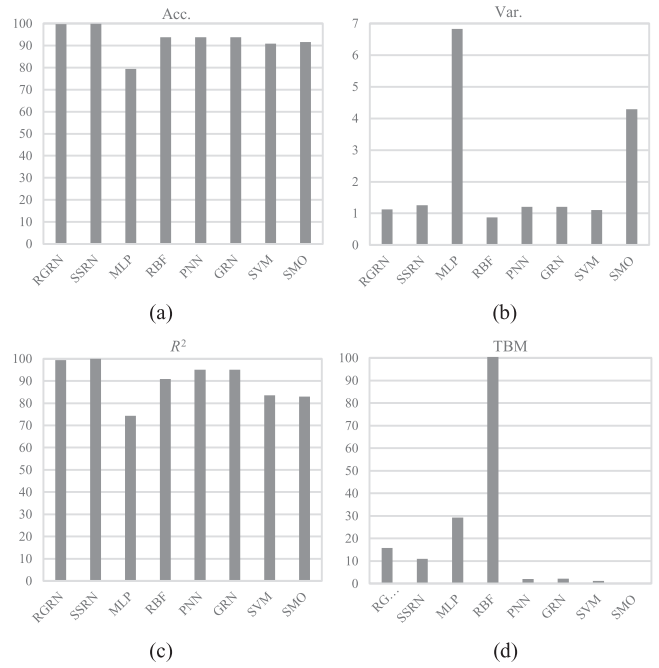

Fig. 8. Models comparisons. (a) Accuracy. (b) Var. (c) R^2. (d) TBM.

TABLE II
ACCURACIES IN DIFFERENT CLASSES

Model	Circle	Cluster	Repetitive	Spot
RGRN	99.99	99.99	99.99	99.38
SSRN	99.99	99.99	99.99	**99.69**
MLP	66.25	65.00	78.75	90.00
RBF	66.67	75.00	99.99	91.67
PNN	91.67	91.67	99.99	99.99
GRN	91.67	91.67	99.99	99.99
SVM	87.50	75.00	99.99	87.50
SMO	87.50	90.00	98.75	90.00

Each model is trained using 320 instances, 80 instances of each defect class.

detection algorithms reported in the literature, where the highest accuracy achieved to the best of our knowledge was about 93% on similarly generated datasets.

Table II shows the classification accuracies of each class of different ML models. As shown in Table II, it is clear that the proposed method's overall predictive performance is far better than all others, with a misclassification results in only one class with probabilities of 0.62 and 0.31 for Methods I and II, respectively.

A. RGRN Experimental Validation

To validate the proposed models, the number of folds was varied from 2 to 14. In theory, increasing the number of folds could reduce model bias by reducing error rate in variance. However, the improvements come at the expense of computational complexity, since the times of rerunning the model will increase by the number of the added folds, as shown in Fig. 9(a)–(d).

In Fig. 10, the number of GRN models is analyzed to find the optimal number of learners (# of classifiers) that results in the best accuracy and the lowest variance. As the number of classifiers increases, the accuracy improves up to a certain

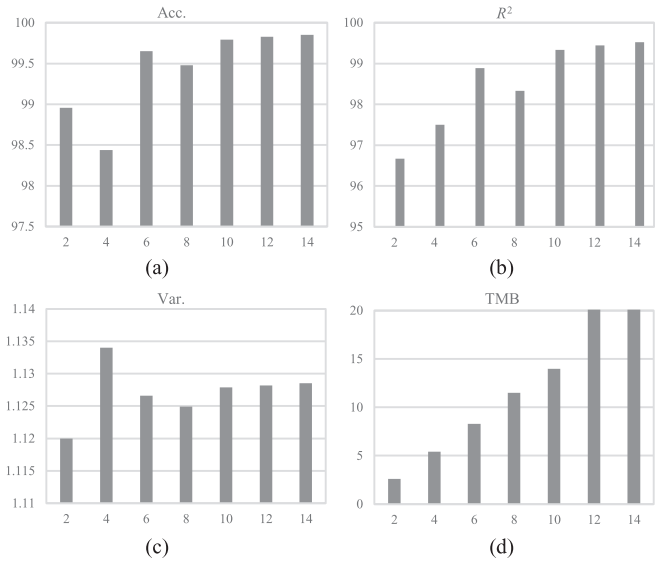

Fig. 9. Model performance comparisons. (a) Accuracy. (b) R^2. (c) Var. (d) TBM in different numbers of folds.

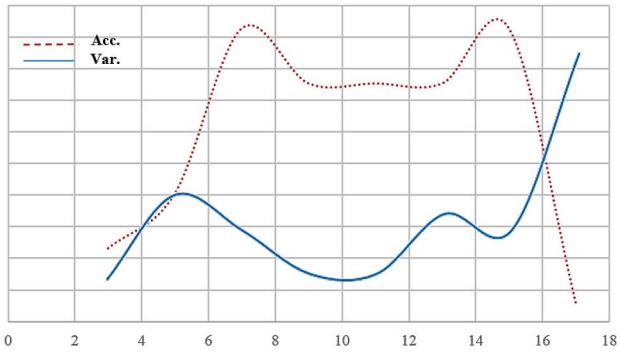

Fig. 10. Number of learners. The effect of changing the number of learners on the accuracy and variance was studied on 12 folds cross-validation. $M = 7$ was chosen as the optimum value.

limit ($M = 7$), then it starts to decrease and stabilize. The same applies for the variance, as M increase its value up to a certain point, then it starts to increase again.

From the previous results, we can conclude that the RGRN algorithm used in Method I is a better classifier than the ones existing in the literature. That is because the novel randomization technique is applied to GRN. The GRN is a fast algorithm that does not require the usage of all training samples to make a prediction of the output. Combining GRN with a powerful randomization technique enables the algorithm to handle complex problems [25], and reduces the misclassification probability significantly. Since, each classifier works independently on a different dataset; hence, the misclassification errors are uncorrelated. Thus, if a classifier makes a wrong decision with probability less than one half, then overall probability of error of the majority voting converges to a very small number as the number of classifiers n increase [31], as illustrated inthe following equation:

$$P_E = \sum_{k=n/2}^{n} p^k (1-p)^{n-k} < \sum_{k=n/2}^{n} \left(\frac{1}{2}\right)^n. \qquad (13)$$

TABLE III
SSRN DATA REDUCTION

SSRN	Fold	Acc.	R^2	σ	TBM
100 (w/o DR)	7	99.702	99.048	1.127	10.755
	8	99.739	99.167	1.127	11.575
	9	99.537	99.213	1.128	14.692
	10	**99.792**	**99.333**	**1.128**	**15.774**
90	7	98.671	97.290	1.282	7.583
	8	98.837	99.288	1.248	8.957
	9	99.742	99.789	1.260	9.364
	10	**99.884**	**99.905**	**1.257**	**10.930**
80	7	99.624	99.699	1.276	6.804
	8	99.671	99.737	1.290	7.864
	9	99.415	98.128	1.270	8.561
	10	**99.737**	**99.157**	**1.277**	**9.630**
70	7	98.740	98.991	1.287	6.218
	8	**99.632**	**98.823**	**1.294**	**7.438**
	9	99.510	99.215	1.283	8.106
	10	99.118	97.881	1.273	9.301
60	7	98.522	97.568	1.268	6.599
	8	99.138	98.227	1.257	6.632
	9	**99.617**	**99.685**	**1.250**	**7.551**
	10	99.483	99.574	1.251	8.800
50	7	98.214	99.048	1.292	5.184
	8	**98.958**	**99.167**	**1.326**	**5.864**
	9	98.148	98.518	1.313	6.706
	10	98.333	98.667	1.295	7.535
40	7	99.624	99.999	1.374	5.000
	8	99.671	99.999	1.374	5.881
	9	**99.708**	**99.999**	**1.374**	**6.468**
	10	99.474	99.596	1.385	7.074
30	7	98.980	99.180	1.338	4.612
	8	98.214	96.414	1.336	5.358
	9	99.206	99.362	1.339	5.908
	10	99.286	99.426	1.339	6.551
20	7	98.571	98.819	1.368	4.403
	8	99.375	99.483	1.355	5.091
	9	**99.444**	**99.541**	**1.359**	**5.569**
	10	97.000	97.521	1.381	6.249
10	7	91.429	83.517	1.271	4.216
	8	92.500	85.577	1.425	4.773
	9	**97.778**	**97.863**	**1.344**	**6.938**
	10	94.000	94.231	1.330	6.418

Furthermore, the over-fitting problem is highly minimized since the classifier is not be able to memorize the data; because no classifier will have full knowledge of the dataset, as each classifier uses its own dataset, only sampled with replacement from the original dataset.

B. SSRN Experimental Validation

To validate the SSRN used in Method II, we first generated nine reduced datasets using Fig. 6, from 90% (include 90% of original data) to 10% then calculated Acc., R^2, σ, and TBM for 7–10 folds for each dataset. Table III shows the comparison of SSRN performance with the original dataset (without reduction) and the reduced datasets: 90%–10%. The best score in each category is underlined, and the best fold scores in each dataset are bolded.

As seen in Table III, as the reduction percentage increases, the TBM decreases, however, with a tradeoff of increasing the variance. The accuracy was improved as compared to the nonreduced dataset when performing 90% reduction on data, however, for all other datasets the accuracies have decreased. Fig. 11 illustrates the performance comparisons of all datasets for 10-fold cross-validation.

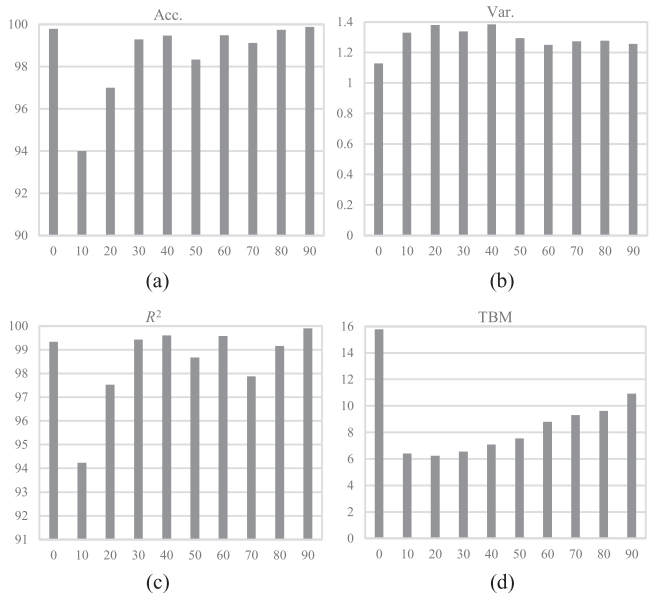

Fig. 11. Performance comparisons on SSRN centroids. (a) Accuracy. (b) Var. (c) R^2. (d) TBM.

TABLE IV
F-MEASURES ON DIFFERENT REDUCTION SCALES

Data	F-Measure
1.0	0.9800
0.9	0.9989
0.8	0.9974
0.7	0.9921
0.6	0.9873
0.5	0.9835
0.4	0.9944
0.3	0.9936
0.2	0.9179
0.1	0.9000

Table IV shows the F-measure value obtained from the classification results on different reduction scales. The F-measure calculates the harmonic mean of the specificity (S_p) and sensitivity (S_n) of the data. It measures that is used as an indication of the quality of the generated clusters by the VD, which can be obtained using the following equation:

$$F_{\text{measure}} = \frac{\left(\beta^2+1\right)\cdot S_p \cdot S_n}{\beta^2 S_p + S_n} \tag{14}$$

$$S_p = \frac{\text{TN}}{\text{TN}+\text{FP}} \tag{15}$$

$$S_n = \frac{\text{TN}}{\text{TN}+\text{FN}} \tag{16}$$

where $\beta = 1$, to give an equal weight to specificity and sensitivity. The closer the F-value was to 1 the better the clustering

quality. From Table IV, it is clear that the clustering performs well, since all the F-value are greater than 0.9.

From the results obtained so far, we can conclude that SSRN used in Method II outperforms the existing models in the literature including RGRN in case of the 90% data reduction. That is because SSRN has all RGRN advantages, while using a reduced dataset minimizing the time complexity as well as generalization ability of the system.

IV. Conclusion

In this paper, we presented two newly developed methods, namely, randomized general regression and simplified subspacing as in Methods I and II, respectively. In Method I, a novel algorithm that combines a general-regression-based consensus learning model with a powerful randomization technique is designed to achieve an accurate and efficient classification of wafer defect patterns. Method I works as a large collection of decorrelated fast general regression learning models, such that each learner depends on the values of a random vector sampled independently and with the same distribution for all learners in the learning network. In addition, due to randomized learning process, the model generalization ability is significantly improved; as memorizing the data is not an option anymore, and the generalization error for Method I converges as the number of learners becomes large. Method II converts input vector $\boldsymbol{x}$ to the centroid vector $\boldsymbol{c}$, ultimately reducing input data significantly. Method II is implemented through a data partitioning and clustering like algorithm creating a final SSRN. *Voronoi*-based data partitioning clusters the original dataset, then the k-means takes the center of each cluster. With SSRN used in Method II, thus, the time complexity of the model is improved significantly. Based on a set of experimental results, we can confidently claim that the utilization of randomized general regression and simplified subspacing techniques as in Methods I and II, for the identification of defect patterns in semiconductor wafers is successful. We believe this is the first time accurate computational classification in such task has been reported achieving detection accuracy above 96%. The proposed techniques in both Methods I and II outperformed all other state-of-the-art ML-based models, in terms of model accuracy and R^2, achieving 99.82% and 99.44%, and 99.88% and 99.91% for Methods I and II, respectively.

Acknowledgment

The authors would like to thank Samsung Electronics and Dr. Jeong for providing the wafer-defect datasets.

References

[1] Semiconductor Industry Association (SIA). (2014). *February Semiconductor Sales Up 11.4 Percent Compared to Last Year* [Online]. Available: http://www.semiconductors.org/news2014/04/04/press_releases_2013/february_semiconductor_sales_up_11.4_percent_compared_to_last_year/

[2] I. M. L. Ferreira and P. J. S. Gil, "Application and performance analysis of neural networks for decision support in conceptual design," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 7701–7708, 2012.

[3] T. Yuan, W. Kuo, and S. J. Bae, "Detection of spatial defect patterns generated in semiconductor fabrication processes," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 3, pp. 392–403, Aug. 2011.

[4] C.-F. Chien, C.-Y. Hsu, and P.-N. Chen, "Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence," *Flex. Serv. Manuf. J.*, vol. 25, no. 3, pp. 367–388, 2013.

[5] C. M. Tan and K. T. Lau, "Automated wafer defect map generation for process yield improvement," in *Proc. 13th IEEE Int. Symp. Integr. Circuits (ISIC)*, 2011, pp. 313–316.

[6] F. L. Chen and S. F. Liu, "A neural-network approach to recognize defect spatial pattern in semiconductor fabrication," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 3, pp. 366–373, Aug. 2000.

[7] G. Choi, S. H. Kim, C. Ha, and S. J. Bae, "Multi-step ART1 algorithm for recognition of defect patterns on semiconductor wafers," *Int. J. Prod. Res.*, vol. 1, pp. 1–14, 2011.

[8] C.-Y. Chang *et al.*, "Wafer defect inspection by neural analysis of region features," *J. Intell. Manuf.*, vol. 22, no. 6, pp. 953–964, 2011.

[9] W. Zeng *et al.*, "A comparison study: Support vector machines for binary classification in machine learning," in *Proc. 4th IEEE Int. Conf. Biomed. Eng. Informat. (BMEI)*, 2011, vol. 3, pp. 1621–1625.

[10] L. Xie, D. Li, and S. J. Simske, "Feature dimensionality reduction for example-based image super-resolution," *J. Pattern Recognit. Res.*, vol. 2, pp. 130–139, 2011.

[11] H. Song *et al.*, "Adaptive virtual support vector machine for reliability analysis of high-dimensional problems," *Struct. Multidiscip. Optim.*, vol. 47, no. 4, pp. 479–491, 2013.

[12] R. Baly and H. Hajj, "Wafer classification using support vector machines," *IEEE Trans. Semicond. Manuf.*, vol. 25, no. 3, pp. 373–383, Aug. 2012.

[13] L. Xie, R. Huang, and Z. Cao, "Detection and classification of defect patterns in optical inspection using support vector machines," in *Intelligent Computing Theories*. New York, NY, USA: Springer, 2013, pp. 376–384.

[14] N. Mallios, E. Papageorgiou, and M. Samarinas, "Comparison of machine learning techniques using the WEKA environment for prostate cancer therapy plan," in *Proc. 20th IEEE Int. Workshops Enabling Technol. Infrastruct. Collab. Enterp. (WETICE)*, 2011, pp. 151–155.

[15] X.-Z. Wang and Ab. B. Musa, "Advances in neural network based learning," *Int. J. Mach. Learn. Cybern.*, vol. 5, no. 1, pp. 1–2, 2014.

[16] C. Dumitru and V. Maria, "Advantages and disadvantages of using neural networks for predictions," *Ovidius Univ. Ann. Ser. Econ. Sci.*, vol. 13, no. 1, pp. 444–449, 2013.

[17] S. Abe, *Support Vector Machines for Pattern Classification*. New York, NY, USA: Springer, 2010.

[18] Y. S. Jeong, S. J. Kim, and M. K. Jeong, "Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 4, pp. 625–637, Nov. 2008.

[19] G. DeNicolao *et al.*, "Unsupervised spatial pattern classification of electrical failures in semiconductor manufacturing," in *Proc. Artif. Neural Netw. Pattern Recog. Workshop*, 2003, pp. 125–131.

[20] K. Machová, F. Barčák, and P. Bednár, "A bagging method using decision trees in the role of base classifiers," *Acta Polytech. Hungarica*, vol. 3, no. 2, pp. 121–132, 2006.

[21] F. Hedayati and E. Kidmose, *Ensemble Bayes Tree*. Berkeley, CA, USA: Electrical Engineering and Computer Science, 2013.

[22] A. J. Ferreira and M. A. T. Figueiredo, "Boosting algorithms: A review of methods, theory, and applications," *Ensemble Machine Learning*. New York, NY, USA: Springer, 2012, pp. 35–85.

[23] H.-C. Kim *et al.*, "Support vector machine ensemble with bagging," in *Pattern Recognition With Support Vector Machines*. New York, NY, USA: Springer, 2002, pp. 397–408.

[24] M. Galar *et al.*, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst. Man Cybern. C., Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[25] K. Joisen *et al.*, "An evaluation of sampling methods for data mining with fuzzy C-means," in *Data Mining for Design and Manufacturing*. Norwell, MA, USA: Kluwer, 2001.

[26] M. De Berg *et al.*, *Computational Geometry*. New York, NY, USA: Springer, 2008.

[27] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[28] M. P. S. Bhatia and D. Khurana, "Experimental study of data clustering using k-means and modified algorithms," *Int. J. Data Mining Knowl. Manage. Process (IJDKP)*, vol. 3, no. 3, pp. 17–30, 2013.

[29] M. Tsujitani and Y. Tanaka, "Cross-validation, bootstrap, and support vector machines," *Adv. Artif. Neural Syst.*, pp. 1–7, 2011.

[30] G. Susto *et al.*, "Prediction of integral type failures in semiconductor manufacturing through classification methods," in *Proc.18th IEEE Conf. Emerging Technol. Factory Autom. (ETFA)*, 2013, pp. 1–4.
[31] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
[32] N. Ye *et al.*, "Optimizing F-measures: A tale of two approaches," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, Edinburgh, U.K., 2012, pp. 289–296.

Fatima Adly received the B.Sc. degree in communications engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2013, where she is currently pursuing the M.Sc. degree.

Her research interests include identifying and classifying defect patterns in semiconductor wafer maps.

Omar Alhussein received the B.Sc. degree in communications engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2013, and the M.A.Sc. degree in engineering science from Simon Fraser University, Burnaby, BC, Canada, in 2015.

From January 2014 to May 2014, he worked as a Research Assistant with Etisalat BT Innovation Centre (EBTIC), Khalifa University. Since May 2014, he has been with the Multimedia Communications Laboratory, Simon Fraser University. His research interests include spanning wireless communications, machine learning, and signal processing.

Mr. Alhussein currently serves as a Reviewer for IEEE COMMUNICATION LETTERS and other flagship conferences.

Paul D. Yoo (M'11–SM'13) received the Ph.D. degree in engineering and information technology from the University of Sydney (USyd), Sydney, NSW, Australia, in 2008.

He was a Postdoctoral Research Fellow with the Centre for Distributed and High Performance Computing, USyd, from 2008 to 2009, and a Doctoral Researcher (Quantitative Analysis) with the Capital Markets Cooperative Research Centre (CRC), administered by the Australia Federal Department for Education, Science, and Training, USyd, from 2004 to 2008. He was with the Advanced Technology Investment Company (ATIC)-Khalifa Semiconductor Research Center and ECE Department., Khalifa University, Abu Dhabi, United Arab Emirates, from 2009 to 2014, as an Assistant Professor of Data Science. He is currently a Lecturer with the Department of Computing and Informatics, Bournemouth University, Poole, U.K. He is also affiliated with USyd, Aristotle University, and KAIST as a Senior Fellowship Scientist. He has over 50 prestigious journal and conference publications.

Dr. Yoo serves as an Editor for IEEE COMMUNICATIONS LETTERS and *Big Data Research*.

Yousof Al-Hammadi received the B.Sc. degree in computer engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2000; the M.Sc. degree in telecommunications engineering from the University of Melbourne, Melbourne, Australia, in 2003; and the Ph.D. degree in computer science and information technology from the University of Nottingham, Nottingham, U.K., in 2009.

He is an Assistant Professor with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi. His research interests include artificial intelligence and pattern recognition with special focus on the development of various detection models in industrial and cyber-security applications.

Kamal Taha (S'09–M'09–SM'14) received the Ph.D. degree in computer science from the University of Texas at Arlington, Arlington, TX, USA, in 2010.

He has been an Assistant Professor with the Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates, since 2010. He has authored over 60 refereed publications that have appeared in prestigious top ranked journals, conference proceedings, and book chapters. He was as an Instructor of Computer Science at the University of Texas at Arlington, from August 2008 to August 2010. He worked as an Engineering Specialist with Seagate Technology, CA, USA, from 1996 to 2005. His research interests include information forensics and security, bioinformatics, information retrieval, data mining, and databases, with an emphasis on making data retrieval and exploration in emerging applications more effective, efficient, and robust.

Dr. Taha currently serves as a Member of the Program Committee, Editorial Board, and Review Panel for a number of international conferences and journals.

Sami Muhaidat (M'08–SM'11) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2006.

From 2007 to 2008, he was an Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON. From 2008 to 2012, he was an Assistant Professor with the School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada. He is currently an Assistant Professor with Khalifa University, Abu Dhabi, United Arab Emirates, and a Visiting Reader with the Faculty of Engineering, University of Surrey, Surrey, U.K. His research interests include advanced digital signal processing techniques for image processing and communications, machine learning, cooperative communications, vehicular communications, Multiple Input Multiple Output (MIMO), and space-time coding. He has authored more than 100 journal and conference papers on these topics.

Dr. Muhaidat currently serves as an Editor for IEEE COMMUNICATIONS LETTERS and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He was the recipient of several scholarships during his undergraduate and graduate studies. He was also a winner of the 2006 NSERC Postdoctoral Fellowship competition.

Young-Seon Jeong received the Ph.D. degree in industrial and systems engineering from Rutgers University, New Brunswick, NJ, USA, in 2011.

He is currently an Assistant Professor with the Department of Industrial Engineering, Chonnam National University, Gwangju, South Korea. His research interests include statistical data mining in the semiconductor manufacturing process, wavelet application for functional data analysis, and intelligent transportation systems.

Uihyoung Lee received the Ph.D. degree in material science and engineering from Seoul National University, Seoul, South Korea, in 2007.

He started working on 3-D integration with Samsung Electronics, Suwon, South Korea, where he has been for more than 7 years and where he is currently a Principal Engineer and Team Representative of the Wafer Level Packaging in Thin Film Team.

Mohammed Ismail (S'80–M'82–SM'84–F'97) received the B.Sc. and M.sc degrees in electrical engineering from Cairo University, Cairo, Egypt., in 1974 and 1978, respectively and the Ph.D. degree in electrical engineering from the University of Manitoba, Winnipeg Manitoba, Canada, in 1983.

He is a Prolific Author and Entrepreneur in the field of chip design and test. He has served on the Faculty of Ohio State University (OSU), ElectroScience Laboratory, Columbus, OH, USA. He held a Research Chair with the Swedish Royal Institute of Technology (KTH), Stockholm, Sweden. He had visiting appointments at Aalto University, Espoo, Finland; North (NTH) and the University of Oslo, Oslo, Norway; Twente University, Enschede, The Netherlands; and the Tokyo Institute of Technology, Tokyo, Japan. He joined Khalifa University (KUSTAR), Abu Dhabi, United Arab Emirates, in 2011, where he was an Advanced Technology Investment Company (ATIC) Professor Chair, the Founding Head of the Electrical and Computer Engineering Department, and the Founding Director of KUSTAR's Semiconductor Research Center. He is also serving as Co-Director of the ATIC-Semiconductor Research Corporation (SRC) Center of Excellence on Energy Efficient Electronic systems (ACE4S), Abu Dhabi. He has served as a Corporate Consultant to over 30 companies. He advised the work of over 50 Ph.D. degree students and of over 100 M.S. degree students. He authored or co-authored over 20 books and over 150 journal publications, and has eight U.S. patents issued and several pending. His research interests include self-healing design techniques for complementary metal oxide semiconductor radio frequency and mm-wave integrated circuit in deep nanometer nodes.

Mr. Ismail is the Founding Editor of *Analog Integrated Circuits and Signal Processing* and serves as the journal's Editor-in-Chief. He has served IEEE in many editorial and administrative capacities. He was the recipient of the U.S. Presidential Young Investigator Award; the Ohio State Lumley Research Award in 1992, 1997, 2002, and 2007; and the U.S. Semiconductor Research Corporation's Inventor Recognition Award twice.