IEEE Transactions on Semiconductor Manufacturing, Vol. 31, Issue 1, February 2018, pp. 156-165
DOI:10.1109/TSM.2017.2768323

Clustering the Dominant Defective Patterns in Semiconductor Wafer Maps

Kamal Taha, *Senior Member, IEEE*, Khaled Salah, *Senior Member, IEEE*, and Paul D. Yoo, *Senior Member, IEEE*

***Abstract*— Identifying defect patterns on wafers is crucial for understanding the root causes and for attributing such patterns to specific steps in the fabrication process. We propose in this paper a system called DDPfinder that clusters the patterns of defective chips on wafers based on their spatial dependence across wafer maps. Such clustering enables the identification of the *dominant* defect patterns. DDPfinder clusters chip defects based on how dominant are their spatial patterns across all wafer maps. A chip defect is considered dominant, if: (1) it has a systematic defect pattern arising from a specific assignable cause, and (2) it displays spatial dependence across a larger number of wafer maps when compared with other defects. The spatial dependence of a chip defect is determined based on the contiguity ratio of the defect pattern across wafer maps. DDPfinder uses the dominant chip defects to serve as seeds for clustering the patterns of defective chips. This clustering procedure allows process engineers to prioritize their investigation of chip defects based on the dominance status of their clusters. It allows them to pay more attention to the ongoing manufacturing processes that caused the dominant defects. We evaluated the quality and performance of DDPfinder by comparing it experimentally with eight existing clustering models. Results showed marked improvement.**

***Index Terms*— Clustering of defective chips, wafer defect patterns, spatial autocorrelation, spatial dependence, wafer map.**

I. Introduction

Thousands of ICs are typically fabricated on a single semiconductor wafer. Each wafer undergoes various processing steps before it is transformed from a plain silicon wafer to one populated with thousands of ICs. During wafer fabrication, thin layers of metals are deposited on the wafer with intervening steps that insert anneal, dopants, and etch patterns. The deposition of these alternating thin layers of metals produce interconnecting vias (i.e., passages) patterned in the deposited layers. IC chips are highly vulnerable to defects in each of the fabrication processing steps. These defects may cause IC chips to completely malfunction. Defect patterns on semiconductor wafers can be classified into two categories [23]: the first is particle related (random clutter) and the second is process related (systematic cluster). Usually, random defects are caused by cleanliness of the clean room (i.e., clean room environment problems). Systematic defects are typically caused by defective equipment, processes, and/or human mistakes. Random defects can be mitigated by expensive equipment overhaul.

To increase product yield, semiconductor manufacturing companies need to understand the root causes of defective chips and to associate them with specific steps in the fabrication process. Thus, it is crucial for semiconductor industry to effectively translate defective clusters on wafer bin map (WBM) into knowledge for the sake of process improvements and yield enhancements. The most important objective of defect identification is the early determination of process problems in order to reduce the number of scrapped chips [7]. Defective chips tend to have specific spatial patterns. Usually, these defects display spatial dependence across wafer maps, which can be traced back to the root causes of the defects. This property has been exploited by dividing defect patterns into groups called clusters. This is because defective chips usually take place in clusters and exhibit systematic patterns [1, 23]. Defects within a cluster are closely similar internally, while sparsely similar with the defects in other clusters.

Many algorithms have been proposed for clustering defects. These algorithms can be categorized into hierarchical-based, neural network-based, partitioning-based, kernel-based clustering, and mixture model-based clustering [14, 30]. However, most of these algorithms: (1) do not consider the *relative* extent of a defect pattern compared to other defect patterns on wafer maps (i.e., the relative extent of a defect pattern d_i is the degree to which d_i is stretched-out to subsume neighboring chips compared to other defects), (2) discount the role of the spatial dependence of defect patterns across wafer maps. As a result, these algorithms do not handle well overlapped and spherical defective patterns.

We propose in this paper a system called **DDPfinder** (**D**ominant **D**efective **P**atterns **F**inder) that clusters defect patterns and overcomes the limitations of most current algorithms outlined above. It overcomes these limitations by clustering patterns of defective chips on wafer map based on their spatial dependence across *all* wafers. Such clustering enables the identification of the *dominant* (i.e., important) defect patterns. More specifically, DDPfinder clusters chip defects based on how dominant are their spatial patterns across wafer maps. A chip defect is considered dominant, if: (1) it has a systematic defect pattern arising from a specific assignable cause, and (2) it displays spatial dependence across a larger number of wafer maps when compared with other defects. In the framework of DDPfinder, a chip defect's spatial dependence is determined based on its contiguity ratio (*CR*) [1] across *all* wafers. A dominant defective cluster represents key characteristics of a specific type of defect pattern that exhibits spatial dependence across a large number of maps.

DDPfinder uses the dominant chip defects to serve as seeds for clustering the patterns of defective chips. This clustering procedure allows process engineers to prioritize their investigation of chip defects based on the dominance

K. Taha and K. Salah are with the Electrical & Computer Engineering Department, Khalifa University, Abu Dhabi, UAE. E-mails: kamal.taha@kustar.ac.ae, khaled.salah@kustar.ac.ae

P. Yoo is with the Centre for Electronic Warfare, Information and Cyber, Cranfield University, Defence Academy of the United Kingdom, Shrivenham, Swindon, SN6 8LA, United Kingdom. E-mail: p.yoo@cranfield.ac.uk

status of their clusters. It allows them to pay more attention to the ongoing manufacturing processes that caused the dominant defects. Non-dominant defective clusters may not signify ongoing manufacturing processes issues (e.g., handling issues). Therefore, considering such defects can waste the time of process engineers and distract them from investigating the important (i.e., dominant) defects. DDPfinder refines the set of defect patterns by keeping only the dominant ones. The main contributions of this paper can be summarized as follows:

- We propose a novel approach that identifies the dominant defect patterns based on their spatial dependence across wafer maps.
- We propose a novel approach that uses the dominant defect patterns to serve as seeds for clustering the patterns of defective chips.
- We perform experimental evaluation and demonstrate the superiority of DDPfinder when compared with other popular existing schemes.

II. Related Work

Most current proposed approaches for detecting spatial defect patterns on semiconductor wafers fall under four broad categories and several subcategories as shown in Fig. 1.

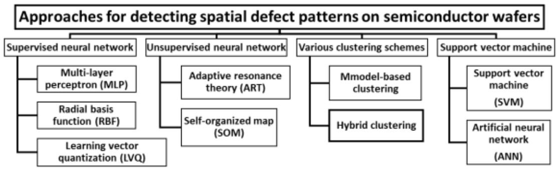

Approaches for detecting spatial defect patterns on semiconductor wafers
Supervised neural network
Unsupervised neural network
Various clustering schemes
Support vector machine
Multi-layer perceptron (MLP)
Radial basis function (RBF)
Learning vector quantization (LVQ)
Adaptive resonance theory (ART)
Self-organized map (SOM)
Mmodel-based clustering
Hybrid clustering
Support vector machine (SVM)
Artificial neural network (ANN)

Fig. 1: The four broad categories and their subcategories for detecting spatial defect patterns on semiconductor wafers

As shown in Fig. 1, the first category includes MLP, RBF, and LVQ approaches. A feed-forwarded neural network (NN) is the most frequently used technique for extracting knowledge from data in problems involve classification and regression [29]. Al Shawish [9] introduced an algorithm that combines a neural regression-network consensus learning model with a randomization technique to classify wafer defect patterns. Adly et al. [10] proposed a framework for identifying defect patterns using simplified subspaced and randomized general regressions as well as Voronoi-based data partitioning for clustering. The main limitation of the supervised neural network approach is its expensive computation time.

As shown in Fig. 1, the second category includes ART and SOM approaches. Lee et al. [16, 17] designed an unsupervised self-organizing map (SOM) algorithm using data sampling methodology. The algorithm clusters the spatial chip locations that have similar defect features. Liu et al. [18] and Chen et al. [5] employed adaptive resonance theory (ART) techniques to detect special types of defect patterns on WBM. Hsu et al. [12] introduced a hybrid method to detect defective patterns by integrating ART network and spatial statistics. Palma et al. [25] adopted SOM and ART as wafer classifiers using extensive simulated and real data sets.

As shown in Fig. 1, the third category includes model-based clustering [6, 31] and hybrid clustering [16] approaches. Chien et al. [6] used Multi-way principal component analysis and data mining techniques to diagnose and monitor the semiconductor fabrication process. Yuan et al. [31] proposed Bayesian model-based clustering algorithms for clustering spatial defective pattern on semiconductor wafers.

As shown in Fig. 1, the fourth category are approaches based on SVM and ANN models. SVM and ANN are the most used models for classifying defect patterns due to their strong versatility and performance [3, 4, 13, 19, 20, 26, 28, 29]. The ANN approaches are: (1) simple, (2) able to handle multi-dimensional problems, and (3) relatively fast [19, 24, 29].

III. Outline of the Approach

In this section, we present an overview of our approach in terms of the sequential processing steps taken by DDPfinder to cluster patterns of defective chips on wafer map. These sequential steps are summarized as follows:

1) ***Grouping chips in wafer maps into Voronoi regions***
   a) *Selecting Sample Chips in Wafer Maps to Represent all Chips in the Wafers:* Let *n* be the number of chips in a wafer. DDPfinder selects *k* sample chips randomly to represent the *n* chips. This process is described in more details in section IV-1.
   b) *Partitioning the Wafer Maps into Voronoi Regions:* The *k* sample chips serve as seeds for constructing *k* Voronoi regions [10]. The *n* chips will be clustered into the *k* Voronoi regions. This process is described in more details in section IV-2.

2) ***Identifying the defective centroid points of Voronoi regions and computing their contiguity ratios:***
   a) *Fetching the Centroid Point of Each Voronoi Region:* In the framework of DDPfinder, each centroid point serves as a representative of its Voronoi region [10]. This will significantly reduce the size of processed data and improves the computation time. This process is described in more details in section V-1.
   b) *Identifying the Defective Centroid Points:* To identify the defective centroid points, DDPfinder first applies a spatial filter to remove outliers and random defects. Then, it identifies each centroid point as either non-defective or defective. DDPfinder considers a centroid point defective, if it resides in a defective Voronoi region. Section V-2 describes this process in details.
   c) *Computing the Contiguity Ratios of the Defective Centroid Points:* DDPfinder determines the spatial autocorrelation of each defective centroid point *cp* by computing the Contiguity Ratio (CR) [1] of *cp* with respect to its neighboring defective centroid points. This process is described in details in section V-3.

3) ***Identifying the dominant defective centroid points:*** To identify the dominant defective centroid points, DDPfinder assigns a score to each candidate defective centroid point cp_i. The score reflects the dominance status of cp_i relative to the other defective centroid points. The score is determined based on the contiguity ratio of cp_i. This process is described in more details in section VI.

4) ***Clustering patterns of chip defects based on their spatial dependence on the dominant defective centroid points:*** DDPfinder clusters defective Voronoi regions based on

how dependent are the spatial patterns of their centroid points on the dominant defective centroid points across wafer maps. That is, DDPfinder uses the dominant centroid points to serve as seeds for clustering the patterns of defective chips. Thus, each cluster will reflect the spatial dependence of its Voronoi regions on a dominant defective centroid point across all wafers. This allows process engineers to prioritize their investigation of chip defects based on the dominance status of their clusters. It allows them to pay more attention to the ongoing manufacturing processes that caused the dominant defects. This process is described in more details in section VII.

IV. Grouping Chips in Wafer Maps into Voronoi Regions

DDPfinder employs an efficient data partitioning scheme for grouping chips into spatial regions, which leads to data reduction. The partitioning scheme is based on Voronoi diagram [10]. Voronoi diagrams are used in many practical applications related to science and technology. A Voronoi region consists of all points closer to a fixed site than any other site. The Voronoi diagram clusters the entire vector space into smaller and manageable Voronoi regions. We present in this section the techniques adopted by DDPfinder for constructing the Voronoi regions, fetching the centroid points of the regions, and identifying the defective ones.

1) Selecting Sample Chips in Wafer Maps to Represent all Chips in the Wafers

Let n be the number of chips in a wafer. For the sake of efficient processing, DDPfinder selects k sample chips randomly to represent the n chips. The k sample chips serve as seeds for constructing k Voronoi regions. That is, each of the k sample chips serves as a base of a Voronoi region. Eventually, the n chips will be clustered into the k Voronoi regions. Then, the centroid point of each Voronoi region is fetched using the K-means algorithm [21]. Each centroid point will serve as a representative of its Voronoi region. That is, the centroid point of a Voronoi region V_x will serve as a representative of all the chips that reside inside V_x. By representing Voronoi regions by their centroid points, the size of processed data will be significantly reduced. This will lead to improving the computation time complexity.

DDPfinder selects the k sample chips in such a way that: (1) the intensity of the selected chips at the wafer's edges is greater than that in the wafer's middle, and (2) the intensity of the selected chips at the wafer's middle is greater than that in the wafer's center. That is, if E, M, and C are the percentages of selected chips at a wafer's edges, middle, and center respectively, then $E\% > M\% > C\%$. This is because: (1) the yield in the near-edge region is usually as much as 50% less than the yield in the center region [2], and (2) the high yield loss in the near-edge region can have a significant impact on the overall wafer yield and fab profit. Since a large wafer's edges and center account for about 23% and 20%, respectively, of the wafer's area [2], we consider the outer 25% area as edges, the inner 25% area as center, and the remaining area as middle.

We present a running example in this section using a rectangular image shapes depicting a small-size wafer map.

Example 1: Fig. 2 shows the defective chips and the selected k sample chips in the small-size wafer map of our running example. The figure shows how the k sample chips are selected in such a way that their intensity at the wafer's edges is greater than that at the middle, and their intensity at the middle is greater than that at the center.

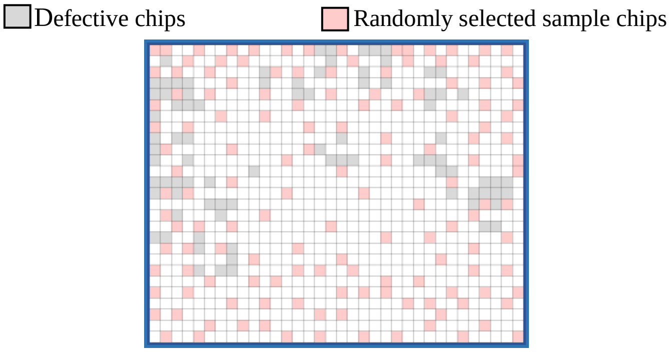


Fig. 2: The defective chips and the selected k sample chips in the small-size wafer of our running example, which represents a reference wafer map.

2) Partitioning the Wafer Maps into Voronoi Regions

A wafer map is partitioned into Voronoi regions. Let V_{ci} denote the Voronoi region that contains the sample chip c_i. V_{ci} contains all chips closer to c_i than any other sample chip. That is, for each of the selected k sample chips there is a corresponding Voronoi region containing all chips closer to this sample chip than to any other sample chip. Thus, the wafer is partitioned into k Voronoi regions. DDPfinder uses the Forgy method [8] for constructing the k Voronoi regions. The method assigns each chip in the wafer to one of the k Voronoi regions. The method employs the K-means algorithm [21] for constructing the k Voronoi regions, whose initial means are the k sample chips. The method constructs a Voronoi region V_{ci} from a set of chips, whose distances to c_i mean produces the least within-V_{ci} sum of squares.

Example 2: Fig. 3 exhibits the partitioning of the wafer map of our running example into Voronoi regions based on the selected k sample chips shown in Fig. 2.

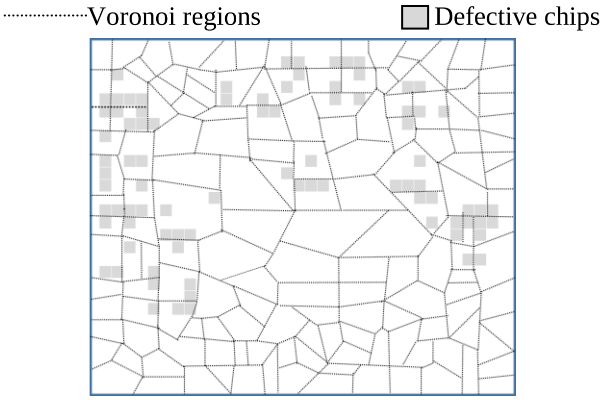


Fig. 3: The partitioning of the wafer map of our running example into Voronoi regions based on the selected k sample chips shown in Fig. 2.

V. Identifying the Defective Centroid Points of Voronoi Regions and Computing their Contiguity Ratios

1) *Fetching the Centroid Point of Each Voronoi Region*

The centroid point of each Voronoi region is fetched using the *K*-means algorithm. This centroid point becomes the new mean of the Voronoi region. By using this technique, the overall time complexity for classifying defects will be reduced significantly, because the centroid point of each Voronoi region will be used as a representative of all chips within the region [15, 21, 22]. This will cause the size of the processed data to be significantly reduced.

Example 3: Fig. 4 shows the centroid points of the voronoi regions in the wafer map of our running example.

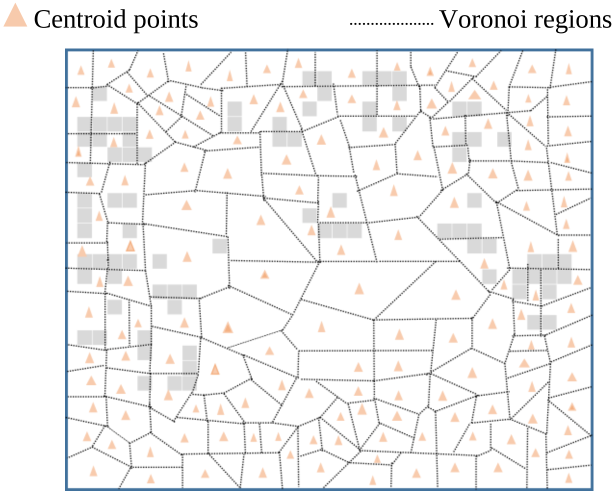


Fig. 4: The centroid points of the voronoi regions in the wafer map of our running example.

2) *Identifying the Defective Centroid Points*

Each centroid point is identified as either non-defective or defective. A centroid point is considered defective, if its Voronoi region is defective. A Voronoi region is considered defective, if the percentage of defective chips within its region exceeds a specific threshold β, and vice versa. Since each Voronoi region *V* is represented by its centroid point *cp*, *cp* is considered defective if *V* is defective and vice versa. In the framework of DDPfinder, each defective centroid point is assigned the value 1 and each non-defective one is assigned the value 0. The value of the threshold β is heuristically determined. In our experimental results, we considered β to be the average percentage of defective chips in the selected *k* sample chips in all reference wafers, as shown in Equation 1:

$$\beta = \frac{1}{mk}\sum_{j=1}^{m}\sum_{i=1}^{k} C_i \qquad (1)$$

- k : Number of sample chips on a wafer.
- m : Number of defective wafers captured by in-line inspection tools during a fabrication processing step.
- $C_i = \begin{cases} 1 & \text{for a defective sample chip} \\ 0 & \text{for a non-defective sample chip} \end{cases}$

Example 4: Fig. 5 shows the defective centroid points of the voronoi regions in the wafer map of our running example.

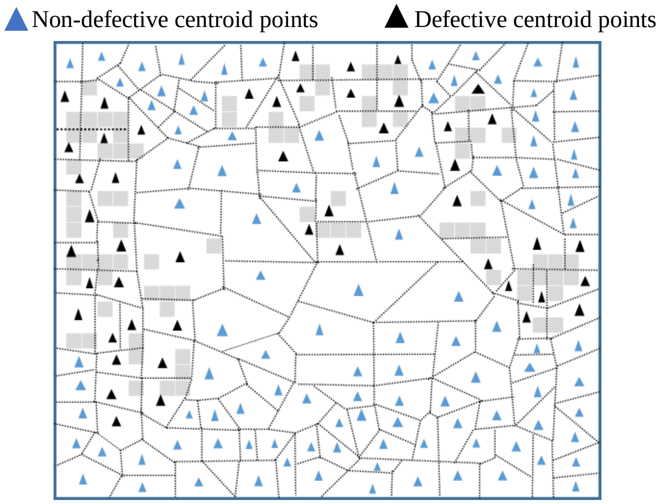


Fig. 5: The defective centroid points of the voronoi regions in the wafer map of our running example.

3) *Computing the Contiguity Ratios of the Defective Centroid Points*

The severity of a defective centroid point can be assessed by how its defect pattern is stretched-out. That is, it can be assessed by the extent of its defect pattern's contiguity to subsume neighboring defective centroid points. Therefore, DDPfinder determines the spatial autocorrelation of each defective centroid point by computing its Contiguity Ratio (*CR*) [1] with regard to its neighboring defective centroid points. *CR* is also known as a measure of spatial autocorrelation. Two centroid points are considered neighbors, if the distance separating them is less than a heuristically determined value. This value is influenced by the size of wafer. A defective centroid point is considered to have high positive spatial autocorrelation, if the value of its *CR* is greater than a predefined threshold δ. We compute *CR* using the formula in Equation 2, which was proposed by Moran [1]:

$$CR = \frac{n\sum_{i\neq j} w_{ij}\left(x_i - \bar{x}\right)\left(x_j - \bar{x}\right)}{2c\sum\left(x_i - \bar{x}\right)^2} \qquad (2)$$

With mathematical manipulations, *CR* can be rewritten as:

$$CR = \left(pc_{00} + qc_{11}\right)/\left(cpq\right) - 1 \qquad (3)$$

c_{00}: Number of functional centroid points with values 0 that are neighboring to an active centroid point under consideration,

c_{11}: Number of defective centroid points with values 1 that are neighboring to an active centroid point.

$p = n_1 / n$,

$q = n_0 / n$, and

n_0, n_1: Numbers of functional and defective chips, respectively, on wafer.

VI. Identifying the Dominant Defective Centroid Points

Most current algorithms that cluster defects do not consider the *relative* extent of each defect pattern compared to other defect patterns on a wafer map. The relative extent of a defect pattern d_i is the degree to which d_i is stretched-out to subsume neighboring chips compared to other defects. In other words, they do not distinguish between dominant and non-dominant defect patterns across all wafers. Non-dominant defect patterns are uninformative and may not signify existing manufacturing processes issues. Therefore, considering such defects can waste the time of process engineers. To overcome this problem, DDPfinder refines the set of defective centroid points by keeping only the dominant ones. It clusters defective centroid points based on how dominant are their spatial patterns across all wafer maps. A defective centroid point is deemed dominant, if: (1) it has a systematic defect pattern arising from a specific assignable cause, and (2) it displays spatial dependence across a larger number of wafer maps when compared with other defects. In the framework of DDPfinder, a chip defect's spatial dependence is determined based on its contiguity ratio (*CR*) [1] across *all* wafers.

To identify the dominant defective centroid points, DDPfinder assigns a score to each candidate defective centroid point cp_i. The score reflects the dominance status of cp_i relative to all other defective centroid points. The score is determined based on the contiguity ratio of cp_i, which is computed using Equation 3 (recall section V-3). Towards this, DDPfinder determines the pairwise *beats* and *looses* for each candidate defective centroid point based on its contiguity ratio.

Let CR_i be the contiguity ratio of the defective centroid point cp_i. Let CR_j be the contiguity ratio of the defective centroid point cp_j. Let n be the number of wafer maps where CR_i is greater than CR_j. Let m be the number of wafer maps where CR_j is greater than CR_i. Candidate defective centroid point cp_i beats candidate defective centroid point cp_j, if n is greater than m. Each candidate defective centroid point cp_x is assigned a score S_{cp_x}, which is the difference between the number of times that cp_x *beats* the other candidate defective centroid points and the number of times it *loses*.

> *Definition 1 – The score of a candidate defective centroid point:* Let CR_i and CR_j be the contiguity ratios of the defective centroid points cp_i and cp_j respectively. Let n be the number of wafer maps where CR_i is greater than CR_j. Let m be the number of wafer maps where CR_j is greater than CR_i. Let $cp_i > cp_j$ denote the case when n is greater than m. Given the dominance relation > on the set V_{cp} of candidate centroid points, the score S_{cp_i} of cp_i equals: $\left|\left\{cp_j \in V_{cp} : cp_i > cp_j\right\}\right| - \left|\left\{cp_j \in V_{cp} : cp_j > cp_i\right\}\right|$.
>
> The following are two properties of this scoring method: (1) the sum of all scores is always zero, and (2) the lowest possible score is $-(|V_{cp}|-1)$ and the highest possible score is $+(|V_{cp}|-1)$.

Let $\hat{S}$ be the absolute value of the largest negative score. We normalize the scores of candidate centroid points by adding $\hat{S}$ to each score and then normalizing the results. The candidate centroid points are ranked based on their dominance scores.

Example 5: Table I shows the contiguity ratios of ten candidate defective centroid points on three dummy wafer maps. Table II shows how the score S_{cp_i} and normalized score $\bar{S}_{cp_i}$ of each candidate defective centroid point cp_i from the ten candidate defective centroid points are computed based on the contiguity ratios of cp_i on the three dummy wafer maps. Consider for example the centroid point cp_1 in Table II. The score S_{cp_1} of cp_1 as shown in the table is -6. This score is computed as follows. As shown in Table I, the number of wafer maps where $cp_1$is greater than cp_2 is 1, while the number of wafer maps where cp_2 is greater than cp_1 is 2. Therefore, cp_1 is beaten by cp_2 and this is denoted by the sign "-" in the entry (cp_2, cp_1) in Table II. As the column cp_1 in Table II shows, cp_1 beat others only one time, it lost to others 7 times, and has equal number of beats and looses 2 times. Therefore, the score S_{cp_1} of cp_1 is -6 (i.e., 1-7 = -6). As Table II shows, the ranks of the ten-candidate defective centroid points based on their normalized scores are as follows: {cp_2, (cp_3, cp_9), (cp_6, cp_8), cp_7, cp_{10}, cp_4, cp_1, cp_5)}.

TABLE I

THE CONTIGUITY RATIOS OF TEN CANDIDATE DEFECTIVE CENTROID POINTS ON THREE DUMMY WAFER MAPS. EACH ENTRY (cp_i, W_j) IS THE CONTIGUITY RATIO OF A CENTROID POINT cp_i ON A WAFER MAP W_j.

	cp_1	cp_2	cp_3	cp_4	cp_5	cp_6	cp_7	cp_8	cp_9	cp_{10}
W_1	+0.29	-0.36	-0.53	0	+0.81	+0.12	+0.29	+0.77	+0.23	+0.87
W_2	-0.12	+0.93	+0.81	+0.29	0	+0.08	+0.49	+0.42	+0.64	+0.35
W_3	+0.38	+0.72	+0.56	+0.17	-0.66	+0.77	-0.26	0	+0.35	-0.45

W and *cp* denote wafer map and centroid point, respectively.

TABLE II

BEATS/LOOSES SCORES OF THE TEN CANDIDATE DEFECTIVE CENTROID POINTS SHOWN IN TABLE I. THE SCORES ARE COMPUTED BASED ON THE CONTIGUITY RATIOS OF THE CENTROID POINTS SHOWN IN TABLE I

	cp_1	cp_2	cp_3	cp_4	cp_5	cp_6	cp_7	cp_8	cp_9	cp_{10}
cp_1	0	+	+	-	+	+	0	+	+	+
cp_2	-	0	-	-	-	+	-	-	-	-
cp_3	-	+	0	-	-	+	-	-	-	-
cp_4	+	+	+	0	-	+	+	-	+	+
cp_5	-	+	+	+	0	+	+	+	+	+
cp_6	-	-	-	-	-	0	+	+	+	+
cp_7	0	+	+	-	-	-	0	+	+	-
cp_8	-	+	+	+	-	-	-	0	+	-
cp_9	-	+	+	-	-	-	-	-	0	-
cp_{10}	-	+	+	-	-	-	+	+	+	0
S_{cp_i}	**-6**	**+7**	**+5**	**-5**	**-7**	**+1**	**0**	**+1**	**+5**	**-1**
$\bar{S}_{cp_i}$	**0.01**	**0.2**	**0.17**	**0.03**	**0**	**0.11**	**0.1**	**0.11**	**0.17**	**0.09**

"+" denotes centroid point cp_i beat centroid point cp_j. "-" denotes centroid point cp_i lost to centroid point cp_j. "0" denotes cp_i and cp_j have equal number of beats and looses. S_{cp_i} and $\bar{S}_{cp_i}$ are the score and normalized score, respectively, for cp_i.

VII. Clustering Patterns of Chip Defects based on their Spatial Dependence on the Dominant Defective Centroid Points

Defective Voronoi regions are clustered based on how dependent are the spatial patterns of their centroid points on the dominant defective centroid points across wafer maps. In particular, DDPfinder uses the dominant centroid points to serve as seeds for clustering the patterns of defective centroid points. Recall that dominant defective centroid points are identified based on their contiguity ratios across all wafer maps, which reflect their spatial patterns' dominance across all the maps. Thus, each cluster will reflect the spatial dependence of its defective Voronoi regions' centroid points on a dominant defective centroid point across all wafer maps. This clustering procedure allows process engineers to prioritize their investigation of chip defects based on the dominance status of their clusters. It allows them to pay more attention to the ongoing manufacturing processes that caused the dominant defects. Non-dominant defective clusters may not signify ongoing manufacturing processes issues (e.g., handling issues). Therefore, considering such defects can waste the time of process engineers and distract them from investigating the important (i.e., dominant) defects.

First, defective centroid points are ranked based on their dominance scores, which are computed as described in section VI. The most dominant defective centroid points are given priority to serve as seeds for constructing clusters. Each cluster consists of:

1) A defective Voronoi region V_x, whose defective centroid point is dominant.

2) Defective Voronoi regions that are neighboring to V_x in the wafer map.

Let L be the list of defective centroid points ranked based on their dominance scores described in section VI. Each cluster R is constructed following these steps:

i. Select the most dominant defective centroid point cp_x in the list L. Include the Voronoi region, whose centroid point is cp_x to cluster R.

ii. Remove cp_x from the list L.

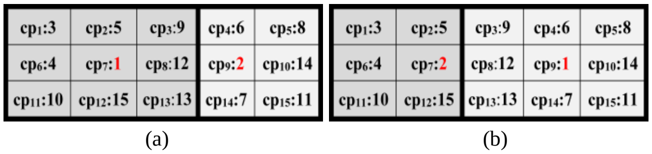
iii. Select the set S_x of Voronoi regions, whose defective centroid points are in the list L and are neighboring to the Voronoi region, whose centroid point is cp_x in the wafer map. Include the set S_x to cluster R.

iv. Remove the centroid points of the set S_x from the list L.

v. Repeat steps i-iv until the list L is exhausted.

For example, consider the following: (1) the defective Voronoi region V_x, whose centroid point is cp_x is neighboring to the Voronoi regions V_y and V_z, whose centroid points are cp_y and cp_z respectively, (2) V_y and V_z are not neighbors, and (3) the dominance ranks of cp_x, cp_y, and cp_z are as follows: $cp_y > cp_z > cp_x$. V_x will be assigned to the cluster containing V_y and not to the cluster containing V_z. Example 6 illustrates this point.

Example 6: Fig. 6 shows how the clustering of Voronoi regions can vary based on the dominance ranks of their centroid points. In the figure, cp_i:x denotes that *x* is the dominance rank of the Voronoi region, whose centroid point is *i*. The ranks of the centroid points in Fig. 6(a) and Fig.6(b) are the same except for cp_7 and cp_9. As can be seen, the clustering changes by swapping the dominance ranks of cp_7 and cp_9.



cp_1:3	cp_2:5	cp_3:9	cp_4:6	cp_5:8
cp_6:4	cp_7:1	cp_8:12	cp_9:2	cp_{10}:14
cp_{11}:10	cp_{12}:15	cp_{13}:13	cp_{14}:7	cp_{15}:11

(a)

cp_1:3	cp_2:5	cp_3:9	cp_4:6	cp_5:8
cp_6:4	cp_7:2	cp_8:12	cp_9:1	cp_{10}:14
cp_{11}:10	cp_{12}:15	cp_{13}:13	cp_{14}:7	cp_{15}:11

(b)

Fig. 6: An illustration of how the clustering of Voronoi regions can vary based on the dominance ranks of their centroid points. The clustering changed by swapping the ranks of centroid points cp_7 and cp_9. Each square represents a Voronoi region and the background colors represent clusters.

VIII. Experimental Results

We implemented DDPfinder in Java, run on Intel(R) Core(TM) i7 processor, with a CPU of 2.70 GHz and 16 GB of RAM, under Windows 10. We evaluated the quality of DDPfinder by comparing it with the following models:

1) The following models, which we previously proposed:
   - Simplified subspaced regression network (SSRN) [10]
   - Randomized general regression neural network (RGRN) [9].
2) The following three widely known and used models:
   - Support vector machines (SVMs).
   - Sequential minimal optimization (SMO).
   - Artificial neural networks (ANNs) including general regression neural (GRN), radial basis function (RBF), probabilistic neural network (PNN), and multi-layer perceptron (MLP).

1) *Compiling a Dataset for the Evaluation*

The dataset used in the experiments is a mixture of real-world wafer maps provided by Samsung Electronics in Korea, and data generated by Jeong *et al.* [32] based on the approach proposed by DeNicolao *et al.* [11]. The dataset includes the most common wafer defect patterns. The following four different defect patterns were generated: spot, circle, repetitive, and cluster. For each defect pattern, a different probabilistic model was used to represent the position of a defective die on the wafer. We used the probabilistic expressions for representing the position of a defective die on the wafer proposed in [11]. 20 × 20 wafer maps were constructed for wafers containing 400 chips each. 80 wafer maps were generated for each defect pattern by adjusting the controlling parameters of its probabilistic model as described above. This allowed us to get variations of the same pattern, such as different sizes, locations, and thicknesses.

2) *Evaluating the Prediction Performance using 10-fold Cross Validation*

The most common and well-accepted statistical method to evaluate the performance of a classifier is cross-validation. Therefore, we use cross-validation to assess the predictive performance of DDPfinder using the following metrics:

- Variance (σ): $\boldsymbol{\sigma}$ provides a good idea about a model's generalization ability and stability. As shown in Equation 4, σ measures the deviation of each data point in the dataset from the mean.
- Coefficient of determination (R^2): R^2 is used as an indication of a model's capability to correctly explain and predict future clustering outcomes. R^2 is measured using Equation 5.
- Clustering accuracy (γ): γ reveals a model's ability to correctly cluster defect patterns. As shown in Equation 6, it is calculated by comparing the predicted clustering output with the actual output.
- *F*-measure: It calculates the harmonic mean of the specificity (*Sp*) and sensitivity (*Sn*) of the result. It is computed as shown in Equation 7.
- Time complexity (TBM). The time required to cluster results

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \qquad (4)$$

where *n* is the size of data, x_i is the actual value of the data, and $\bar{x}$ is the mean.

$$R^2 = 100 \times \left(1 - \frac{\sum_{i=1}^{n}(x_i - m_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right) \qquad (5)$$

where m_i is the predicted output

$$\gamma = \frac{\text{length}\left(X = \hat{x}\right)}{\text{length}\left(X\right)} \qquad (6)$$

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

$$F_{measure} = \frac{\left(\beta^2+1\right).S_p\,.\,S_n}{\beta^2 S_p + S_n} \qquad (7)$$

where, $S_p = \frac{TN}{TN+FP}$, $S_n = \frac{TN}{TN+FN}$, and $\beta = 1$.

To assess the predictive performance of the models, we performed 10-fold cross-validation. The dataset is partitioned (at random) into 10 disjoint subsets. The models are evaluated ten times, where at each time a different subset of the data is used for testing while the remaining nine subsets are used for training the models. DDPfinder uses the training dataset to identify the dominant defective centroid points. Then, it clusters the patterns of chip defects on test wafers based on the dominant defective centroid points identified from the training dataset. The 80 wafer maps generated for each defect pattern are used as ground truth data. Fig. 7 shows the overall average γ, σ, R^2, and TBM, respectively, for each model.

Table III shows the clustering performance of the different models under folds 7-10. In theory, increasing the number of folds could reduce a model's bias by reducing error rate in variance. However, the improvements come at the expense of computational time complexity, since the times of rerunning the model will increase by the number of added folds.

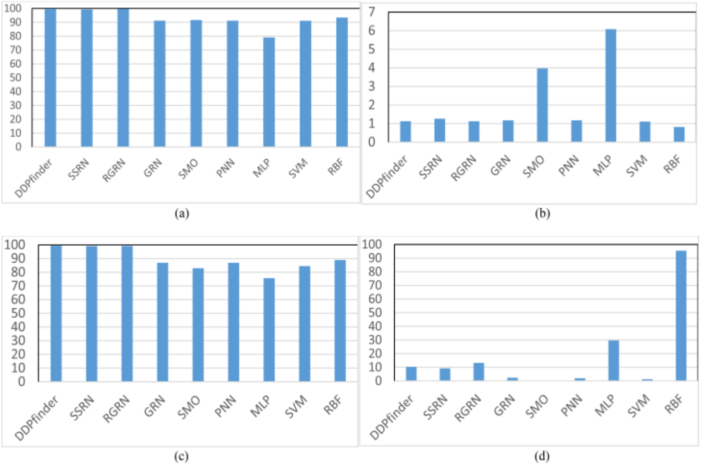


Fig. 7: Overall average of: (a) γ, (b) σ , (c) R^2, and (d) TBM (in seconds).

TABLE III
CLUSTERING PERFORMANCE OF THE MODELS UNDER FOLDS 7-10

Model	Fold	γ	R^2	σ	TBM (seconds)
DDPfinder	7	99.786	99.250	1.114	8.204
	8	99.873	99.329	1.073	9.582
	9	99.835	99.428	1.213	11.410
	10	99.803	99.776	1.101	12.356
SSRN	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
RGRN	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
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
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
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
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
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
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

The following are our observations regarding the results:

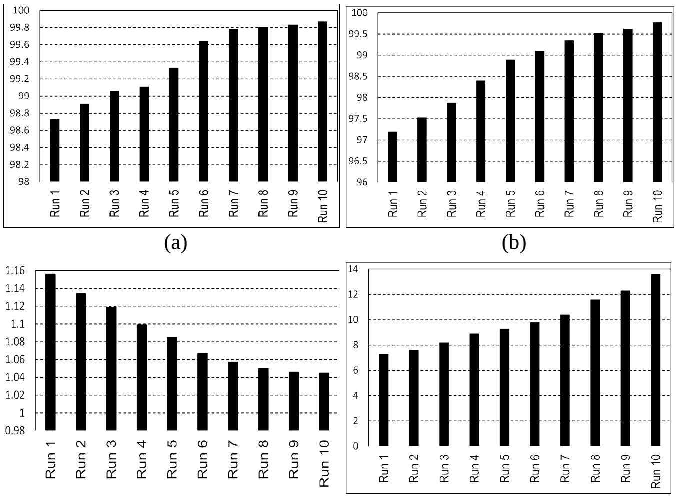
- ***Accuracy (γ):*** As Fig. 7-a and Table III show, DDPfinder outperformed the other eight models in terms of accuracy. However, the performance of DDPfinder over RGRN was slight, where the average γ of DDPfinder and RGRN were 99.824 and 99.692, respectively.

- ***Variance (σ):*** As Fig. 7-b and Table III show, DDPfinder outperformed SSRN, RGRN, GRN, SMO, PNN, and MLP. However, SVM and RBF outperformed DDPfinder.
- ***Capability to predict future outcomes correctly (R^2):*** As Fig. 7-c and Table III show, DDPfinder outperformed the other eight models. However, its performance over RGRN was slight, where the average R^2 of DDPfinder and RGRN were 99.446 and 99.190 respectively.
- ***Time complexity (TBM):*** As Fig. 7-d and Table III show, DDPfinder outperformed only RGRN, MLP, and RBF, while the remaining models outperformed DDPfinder. Therefore, we need to investigate approaches for improving DDPfinder's time complexity in a future work.

3) Evaluating the Prediction Performance using Cumulative-Validation Dataset

Wafer defect data in real-world accumulates over time and such data abundance should be utilized to enhance defect prediction accuracy. Therefore, every time defect data is collected from a set of recently fabricated wafers, DDPfinder updates and optimizes the current beats/looses scores of defective centroid points (recall Table II) based on this recently collected defect data. In this section, we aim at determining whether the prediction performance of DDPfinder improves constantly over time as the size of training dataset increases. That is, we aim at assessing the impact of the increasing size of training dataset on the prediction performance of DDPfinder.

We partitioned the dataset at random into training and testing disjoint subsets. We then performed 10 evaluation runs. The set of training dataset accumulates in each run successively. After each run, the current set of *testing* data will be added to the current set of *training* data, and the accumulating set will be used for training DDPfinder in the next run (i.e., the set of training data is the cumulative of the training and testing data of all previous runs). Fig. 8 shows the results using the metrics shown in Equation 4-7 in addition to TBM. Fig. 9 shows the *F*-measure in each of the 10 runs using the metric shown in Equation 4.



(a) (b)

(c) (d)

Fig. 8: The prediction performance of DDPfinder using cumulative-validation dataset. (a) Accuracy, (b) R^2, (c) Variance, (d) TBM.

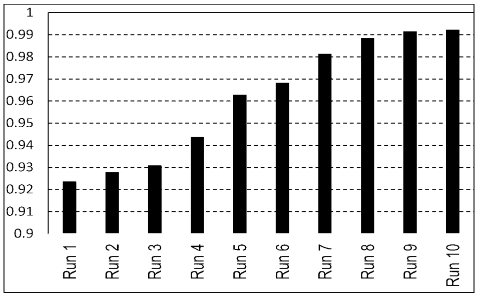


Fig. 9: F-measure on the 10 cumulative rums

As exhibited in Fig. 8 that the prediction performance of DDPfinder improves constantly as the size of training data increases. This is because DDPfinder updates and optimizes the current beats/looses scores of defective centroid points after each run (recall Table II). It does so by considering the Contiguity Ratios of the defective centroid points in the current set of recently fabricated wafers (recall Table I). Thus, clustering accuracy keeps improving over time. As for the *F*-values shown in Fig. 9, they indicate that DDPfinder performs well, since all these values are greater than 0.92. However, as the training dataset increases, DDPfinder's time complexity increases (see Fig. 8-d). These increases are insignificant and justifiable.

Acknowledgment

We would like to thank Samsung Electronics in Korea and Dr. Jeong for providing the real-world wafer maps.

IX. Conclusion

We introduced a system called DDPfinder that clusters the patterns of defective chips on wafer maps and overcomes the limitations of exiting popular algorithms that cluster chip defect. It does so by clustering the patterns of defective chips based on their spatial dependence across *all* wafer maps. This clustering procedure enables the identification of the most *dominant* defect patterns on a wafer map. This allows process engineers to prioritize their investigation of chip defects and to pay more attention to the ongoing manufacturing processes that caused the dominant defects. We evaluated the quality and performance of DDPfinder by comparing it with eight models. Results showed marked improvement.

References

[1] A. D. Cliff and J. K. Ord, *Spatial Processes: Models &Applications*. New York: Pion, 1981.

[2] Burkeen et al., Visualizing the Wafer's Edge, KLA-Tencor Yield Management Solutions Magazine, Spring 2007, pp. 18-20.

[3] Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. Journal of Machine Learning Research, 2, 125–137.

[4] C.-Y. Chang, C.-H. Li, Y.-C. Chang, and M. Jeng, "Wafer defect inspection by neural analysis of region features," *J. Intell. Manuf.,* vol. 22, no. 6, pp. 953–964, 2011.

[5] Chen, F. L., & Liu, S. F. (2000). A neural-network approach to recognize defect spatial pattern in semiconductor fabrication. IEEE Transactions on Semiconductor Manufacturing, 13, 366–372.

[6] 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.

[7] Chou, P. B., Rao, A. R., Sturenbecker, M. C., Wu, F. Y., & Brecher, V. H. (1997). Automatic defect classification for semiconductor manufacturing. Machine Vision and Applications, 9, 201–214.

[8] E. Forgy, Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classification, Biometrics 21 (1965) 768.

[9] F. Adly, P. Yoo, S. Muhaidat, Y. Al-Hammadi, U. Lee, and M. Ismail, "Randomized General Regression Network for Identification of Defect Patterns in Semiconductor Wafer Maps". *IEEE Transactions on Semiconductor Manufacturing*, 28(2): 145-152 (2015).

[10] F. Adly, O. Al Hussein, Paul Yoo, Y. Al-Hammadi, Kamal Taha, Sami Muhaidat, Uihyoung Lee, and Mohammed Ismail, "Simplified Subspaced Regression Network for Identification of Defect Patterns in Semiconductor Wafer Maps". *IEEE Transactions on Industrial Informatics*, 11(6): 1267-1276 (2015).

[11] 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.

[12] Hsu, S. C., & Chien, C. F. (2007). Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. Journal of Production Economics, 107.

[13] H. Song, K. K. Choi, I. Lee, L. Zhao, and D. Lamb, "Adaptive virtual support vector machine for reliability analysis of high-dimensional problems," *Struct. Multidiscipl. Optim.*, 47(4), pp. 479–491, 2013.

[14] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. ACM Computing Survey, 3(3), 264–323.

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

[16] Lee, J. H., Yu, S. J., & Park, S. C. (2001a). Design of intelligent data sampling methodology based on data mining. IEEE Transactions on Robotics and Automation, 5(17), 637–649.

[17] Lee, J. H., Yu, S. J., & Park, S. C. (2001b). A new intelligent SOFM-based sampling plan for advanced process control. Expert Systems with Applications, 20, 133–151.

[18] Liu, S. F., Chen, F. L., & Lu, W. B. (2002). Wafer bin map recognition using a neural network approach. International Journal of Production Research, 40, 2207–2223.

[19] L. Xie, R. Huang, and Z. Cao, "Detection and classification of defect patterns in optical inspection using support vector machines," in *Intelligent Computing Theories*. Berlin, Germany: Springer, 2013.

[20] L. Xie, D. Li, and S. J. Simske, "Feature dimensionality reduction for example-based image super-resolution," *J. Pattern Recognit. Res.*, vol. 6, no. 2, pp. 130–139, 2011.

[21] 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.

[22] M. De Berg *et al.*, *Computational Geometry*, Springer, 2008.

[23] M. H. Hansen, V. N. Nair, and D. J. Friedman, "Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects," *Technometrics*, vol. 39, no. 3, pp. 241–253, 1997.

[24] 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 Enabl. Technol. Infrastruct. Coll. Enterp. (WETICE)*, Paris, France, 2011.

[25] R. Baly and H. Hajj, "Wafer classification using support vector machines," *IEEE Trans. Semicond. Manuf.*, 25(3), pp. 373–383, 2012.

[26] Su, C. T., Yang, T., & Ke, C. M. (2002). A neural-network for semiconductor wafer post-sawing inspection. IEEE Transactions on Semiconductor Manufacturing, 15, 260–266.

[27] T. C. Bailey and A. C. Gatrell, *Interactive Spatial Data Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[28] W. Zeng, J. Jia, Z. Zheng, C. Xie, and L. Guo, "A comparison study: Support vector machines for binary classification in machine learning," in *Proc. 4th IEEE Int. Conf. Biomed. Eng. Informat. (BMEI)*, vol. 3. Shanghai, China, 2011, pp. 1621–1625.

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

[30] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3), 645–678.

[31] Yuan, T., & Kuo, W. Spatial defect recognition on semiconductor wafers using model-based clustering and Bayesian inference. European Journal of Operational Research, 2008, 190(1), pp. 228–240.

[32] 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.

Kamal Taha is an Associate Professor in the Department of Electrical and Computer Engineering at Khalifa University, UAE, since 2010. He received his Ph.D. in Computer Science from the University of Texas at Arlington, USA, in March 2010. He has over 70 refereed publications that have appeared in prestigious top ranked journals, conference proceedings, and book chapters. Over 20 of his publications have appeared in IEEE Transactions journals. He was as an Instructor of Computer Science at the University of Texas at Arlington, USA, from August 2008 to August 2010. He worked as Engineering Specialist for Seagate Technology, USA, from 1996 to 2005 *(Seagate is a leading computer disc drive manufacturer in the US)*. His research interests span bioinformatics, Information Forensics & Security, information retrieval, data mining, and databases, with an emphasis on making data retrieval and exploration in emerging applications more effective, efficient, and robust. He serves as a member of the Program Committee, editorial board, and review panel for a number of international conferences and journals, some of which are IEEE and ACM journals. He is a Senior Member of the IEEE.

Khaled Salah is a full professor at the ECE Department, Khalifa University of Science, Technology and Research, UAE. He joined the ECE Department in September 2010. Prior to joining Khalifa University, Khaled was with the Department of Information and Computer Science, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. Khaled has over 160 publications related to the areas of cloud computing, big data, and cybersecurity. Khaled was the recipient of Khalifa University Outstanding Research Award 2014, KFUPM University Excellence in Research Award 2009, and KFUPM Best Research Project Award 2010. Khaled will serve as the Track Chair of IEEE Globecom 2018 on Cloud Computing.

Paul D. Yoo is currently with the Cranfield Defence Security, based at the Ministry of Defence establishment, United Kingdom. Prior to this, Dr Yoo held academic/research posts in Sydney, Bournemouth, and the UAE. Dr Yoo serves as Editor of IEEE COMML and Elsevier JBDR journals and holds over 60 prestigious journal and conference publications. Dr Yoo is affiliated with University of Sydney and Korea Advanced Institute of Science and Technology (KAIST) as Visiting Professor. He is a Senior Member of IEEE and a Member of BCS. His research focuses on large-scale data analytics including design and development of computational models and algorithms inspired by intelligence found in physical, chemical and biological systems, and to solve practical problems in security and digital forensics.