A Voting Ensemble Classifier for Wafer Map Defect Patterns Identification in Semiconductor Manufacturing

Muhammad Saqlain

IEEE Transactions on Semiconductor Manufacturing

A Voting Ensemble Classifier for Wafer Map Defect Patterns Identification in Semiconductor Manufacturing

Muhammad Saqlain, Bilguun Jargalsaikhan, and Jong Yun Lee

***Abstract*—A wafer map contains a graphical representation of the locations about defect pattern on the semiconductor wafer, which can provide useful information for quality engineers. Various defect patterns occur due to increasing wafer sizes and decreasing features sizes, which makes it very complex and unreliable process to identify them. In this paper, we propose a voting ensemble classifier with multi-types features to identify wafer map defect patterns in semiconductor manufacturing. Our research contents can be summarized as follows. First, three distinctive features such as density-, geometry-, and radon-based features were extracted from raw wafer images. Then, we applied four machine learning classifiers, namely logistic regression (LR), random forests (RFs), gradient boosting machine (GBM), and artificial neural network (ANN), and trained them using extracted features of original data set. Then their results were combined with a soft voting ensemble (SVE) technique which assigns higher weights to the classifiers with respect to their prediction accuracy. Consequently, we got performance measures with accuracy, precision, recall, *F*-measure, and AUC score of 95.8616%, 96.9326%, 96.9326%, 96.7124%, and 99.9114%, respectively. These results show that the SVE classifier with proposed multi-types features outperformed regular machine learning-based classifiers for wafer maps defect detection.**

***Index Terms*—Feature extraction, manufacturing automation, pattern classification, semiconductor defects.**

I. Introduction

TECHNOLOGY enhancement in semiconductor manufacturing has benefited the companies in term of yield improvement, on-time delivery, cost reduction, cycle-time reduction, and product performance. However, the high operation frequency and power supply troubleshooting have resulted in increased semiconductor defects rate. These defects not only may cause of quality variation, but also create lower product yield and reliability problems. A wafer map (WM) is a collection of visual data about the physical parameter that are collected from semiconductor wafers [1]. It contains basic information about thickness, size, and location of defects on wafers. The WM defects are so common that these cannot be avoided even in the presence of precisely positioned and latest equipment using an almost particle-free environment [2].

Normally, two types of wafer defects occur; local defects and global defects [3]. Local defects are inspected by some significant patterns on wafers, and these are because of some specific reasons such as particles from apparatus, chemical stains, or human mistakes. On the other hand, global defects do not contain any specific patterns and scattered all over the wafer, so it becomes very difficult to find their causes. Some typical local defects are *Center*, *Rings*, *Scratches*, *Donut*, *Semicircle*, *Local*, *Random*, and *Near-full* [4]. These defects contain crucial information about the actual causes of their happening at the manufacturing level. So, the research methodologies about WM defects detection and recognition of spatial patterns are in high demand. These methods can be further used for early prevention of defects by diagnosing their root causes and to enhance the product quality by improving the reliability of the manufacturing system.

There is a need for effective and efficient wafer identification and analysis tools [5]. One way of WM defects detection is to recognize the visual details by experienced semiconductor engineers. They are responsible for selecting random samples from whole wafer data and detect the actual defects causes by analyzing the physical measurement like color, size, and location of defects with high-resolution microscopes. However, this procedure is very costly, inefficient, and time-consuming for huge wafer data set. The defects detection accuracy of human-expert methods is only 45% or lower, which is not acceptable on large scale [6]. Moreover, the ubiquitous use of mobile and embedded devices has increased wafer production capacity. Therefore, most of the wafer manufacturing companies are finding other approaches by evolving into automatic WM defects identification using various machine learning (ML) models.

Generally, ML classifiers for wafer map defect pattern identification (WMDPI) are classified into two categories: supervised and unsupervised. Supervised learning models are used when the class labels for wafers are available and models classify the given unknown data s into different known classes according to the knowledge attained from the previously available training data set. Data classification is a good example of this approach. Artificial neural

Manuscript received February 19, 2019; revised March 5, 2019; accepted March 7, 2019. Date of publication March 11, 2019; date of current version May 3, 2019. This work was supported in part by the Korea Institute for Advancement of Technology grant funded by the Korea Government Ministry of Trade Industry and Energy under Grant N0002429, and in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education under Grant 2017R1D1A1A02018718. This work was financially supported by the Research Year of Chungbuk National University in 2016. *(Corresponding author: Jong Yun Lee.)*

The authors are with the Department of Computer Science, Chungbuk National University, Cheongju 28644, South Korea (e-mail: jongyun@chungbuk.ac.kr).

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/TSM.2019.2904306

network (ANN) [7], support vector machine (SVM) [4], adaptive resonance theory (ART) neural network [8], general regression neural network (GRNN) [5], and back-propagation network (BPN) [9] are commonly used supervised learning algorithms for WM defects identification. Contrarily, unsupervised learning models are used to classify the wafer data where the class labels are unknown. Data clustering is an example of this approach. In clustering models, the number of clusters and their class labels are not defined, so it has advantages over the classification models when some new WM defect patterns are identified during a continuous fabrication process and need to be added into new classes. Some important unsupervised learning models for WM defects clustering are particle swarm optimization (PSO) [8], multi-step ART1 [10], k-means [11], and self-organizing maps (SOM) [12]. Recently, a well-known deep learning model convolutional neural network (CNN) is being applied for WM defects classification and finds state-of-the-art accuracy [42]. It is an end-to-end classifier and does not require manual feature extraction.

However, we can summarize some challenging issues of previous works as follows. First, with the advancement of wafer manufacturing technology, many researchers have investigated WM defect patterns identification methods in recent years, but most of them have used raw wafer data for this purpose. Analyzing large-scale raw wafer data may cause various problems such as expensive storage, computational cost, and lower accuracy. Second, all ML algorithms are suffering from numerous limitations and no individual algorithm is suitable for the entire problem. For example, ANN always needs to retrain the whole network when the new training data set is to be added and SVM can take a long time and involve high computational power for a large data set. Third, due to complex and deep structure, CNN requires comparatively large-scale image data, a lot of hyperparameters tuning, high computational power (such as parallel GPUs), and long training time [43].

So, the use of multi-types features with an ensemble classifier can be a satisfactory solution to these problems. The extracted features may contain valuable hidden information generated from the original WM data. Many studies are focusing on feature extraction techniques for WMDPI, e.g., geometry-based features [4], radon-based features [13], representative features [14], density-based features [15], and texture features [16]. These features are more feasible because they need minimal storage and computation while containing hidden patterns which improve wafer defects identification ability for large scale wafer data. On the other hand, ensemble classification methods have become a popular topic in the field of ML and being used to overcome the limitation of individual classifiers [17]. The goal of ensemble methods is to combine the prediction results of various ML models within given learning parameters and generate a final prediction result to improve the accuracy. Comparing the performance of individual classifiers, ensemble classifiers have shown more effective results regarding stability and robustness [18].

All the ensemble systems consist of three basic pillars, including diversity, training of all member classifiers of ensemble systems, and combining results of all these members using simple or weighted majority voting to get an aggregate result [19]. The performance of ensemble systems depends on the accuracy of the individual classifiers, and the number of base classifiers included (i.e., the more classifiers we included the better ensemble system will perform). However, selection of appropriate classifiers for designing an ensemble system remains a very hard topic. Moreover, all ML classifiers cannot be the best performer to recognize all defect classes. They can be experts in some part of the feature space, and an ensemble model assigns specific weights to the individual classifiers with respect to their expertise. For instance, one of the classifiers can be good to identify *Center* class whereas others can be good for *Donut* class. But the ensemble model of these classifiers will perform accurately to identify both defect classes.

Therefore, this paper proposes a voting ensemble classifier with multi-types features to identify wafer map defect patterns in semiconductor manufacturing. We used a WM-811K data set which is the largest publicly available wafers data and consists of 811,457-real wafers. The detailed contents of our research can be summarized as follows. First, we extracted three sets of valuable features from each input wafer image such as density-based, radon-based, and geometry-based features. These features were combined for an equal contribution to train all base classifiers. Second, we implemented four different state-of-the-art base classifiers namely, logistic regression (LR), random forests (RF), gradient boosting machine (GBM), and artificial neural network (ANN) to train with combined multi-types features of original data set. Then we used weighted averaging or soft voting ensemble (SVE) approach which assigns specific weights to the actual continuous outputs of base classifiers with respect to their prediction accuracy. Lastly, we applied different performance measures such as accuracy, precision, recall, F-measure, and area under the ROC curve (AUC) to get final classification results and evaluate the proposed method. Consequently, after applying numerous parameter settings, the SVE classifier outperformed all individual classification algorithms as well as previously proposed wafer map failure pattern recognition (WMFPR) [4] method and CNN model [44]. The experimental results show the importance of the proposed ensemble classification method with multi-types features for WMDPI.

The rest of this paper is organized as follows: Section II contains related studies and Section III contains research methodology followed by this study. Proposed ensemble approach is described in Section IV. The evaluation results of the proposed method and performance comparison are presented in Section V. The whole study is concluded in Section VI.

II. Related Work

Wafer map defect identification is very crucial for yield enhancement and on-time delivery in the rapidly growing semiconductor manufacturing industry. For this purpose, various ML applications are being applied. Wafer-based clustering approaches are used when the WMs don't contain the class labels. Adaptive resonance theory (ART1) is a commonly used unsupervised learning approach for wafer bin map (WBM) detection [21]. The WBM is a two-dimensional defect pattern generated in the semiconductor fabrication process and is further classified into mixed, random, and systematic failure patterns. Chien *et al.* [11] developed a data mining and

knowledge discovery framework consists of k-mean clustering and Kruskal-Wallis test to analyze a huge amount of wafer manufacturing data and to diagnose failure patterns and fabrication process variations. Hsu [8] proposed a clustering ensemble approach to efficiently identify systematic failure patterns in wafer production. He used particle swarm optimizer (PSO) and k-mean clustering algorithms to create diversity partitions and different label classes and then adaptive response theory network was applied to get the integrated result. Liu and Chien [22] proposed a knowledge-based intelligent system for detection of WM defects by combining graphical user interface (GUI), knowledge database, and wafer clustering solution for yield improvement in wafer manufacturing. Specifically, the clustering solution was integrated with spatial statistics test, adaptive resonance theory (ART) neural network, moment invariant (MI), and cellular neural network for effectively failure pattern detection.

Wu *et al.* [4] presented a wafer map failure pattern recognition (WMFPR) by combining two sets of extracted features such as geometry-based and radon-based and applied the SVM classifier to identify the defect patterns. The selected features were also feasible to find similarity ranking in huge wafer data set of WM-811K. Jeong *et al.* [23] proposed a new methodology in which spatial correlogram is used for the detection of the presence of spatial autocorrelations and for the classification of defect patterns on the WM. Yu and Lu [7] modeled a manifold learning-based WM defect detection and recognition system. In this system, a joint local and nonlocal linear discriminant analysis (JLNDA) was proposed to discover intrinsic manifold information that provides the discriminant characteristics of the defect patterns. Baly and Hajj [5] suggested SVM classifier due to its ability for efficient classification of multi-modal, multivariate, and inseparable wafer data points. Their proposed model applied multidimensional hyperplanes for separating and classifying wafer data into high-yield and low-yield classes.

Recently, a deep learning (DL) model convolutional neural networks (CNN) has become a standard method for any image classification and patterns recognition tasks [43]. It does not require manual feature extraction due to its ability to automatically feature detection from raw image data. A CNN based WM defect pattern classification and image retrieval method was proposed in [42]. It theocratically generated 28,600 WM images for 22 defect classes and improved the test accuracy up to 98.2%. But it used only simulated data for training and validation of the CNN model because real data was highly imbalanced. Another study proposed by Kyeong and Kim [44] used the CNN model for classification of multiple defects on same wafer. To do this, they proposed individual CNN classifiers for each defect class. For example, four CNN models were built to detect *Circle*, *Scratch*, *Ring*, and *Zone* defect classes. As CNN classifier requires a large amount of training data so they used real as well as simulated images for training the CNN models.

Most of the previous literature either used raw wafer images, only one, or two selective features sets. However, we proposed three types of feature sets such as density-based, radon-based, and geometry-based features from raw wafer images. These combined features will ultimately improve the performance of ML classifiers for WMDPI. We also applied multiple classifiers or ensemble model to overcome the limitations of the individual classifiers. An ensemble model combines prediction results from various ML base classifiers to produce a final prediction result. These systems have also many other ML specific application such as feature selection, data fusion, confidence estimation, incremental learning, and addressing imbalance data [24]. Research in ensemble models has been exploded from last two decades and different ensemble-based algorithms has been appeared, namely bagging [25], boosting [26], AdaBoost [27], random forest (combination of decision trees) [28], stacked generalization [20], simple or weighted majority voting [17], consensus aggregation [29], classifier ensemble [30], mixture of experts (MoE) [31], classifier fusion [32], among many others. All these algorithms are varied from each other based on the typical method used for generating ensemble models, the selection of training data for each member classifier, and the combination rule generating the ensemble result. Zhang and Ma [19] described some popular combination rules of various ML classifiers, such as voting based techniques, an algebraic combination of outputs, decision templates, and behavior knowledge space. Saha and Ekbal [17] proposed a voting classifier ensemble method by combing seven diverse ML models, namely decision trees (DT), naive bayes (NB), memory-based learner (MBL), support vector machine (SVM), hidden markov model (HMM), conditional random field (CRF), and maximum entropy (ME) for named entity recognition (NER) system. The results of their study represented that the proposed ensemble classifier outperformed all the individual classifiers.

A voting ensemble method has two flavors (1) majority or hard voting and (2) weighted majority or soft voting. The hard-voting ensemble has also three different versions, where the ensemble system selects the class (a) on which all base classifiers agree (unanimity); (b) classified by more than half of base classifiers (majority); or (c) get the maximum number of votes (plurality). It combines the classifiers outputs using *Jury Theorem*, whose details can be found in [45]. It is not the best option when: (i) the probability of base classifiers for selecting the true class is p for any instance $\mathbf{x}$; (ii) we have an odd number of base classifiers for the binary class problem; and (iii) the outputs of classifiers are independent. Whereas, soft voting ensemble (SVE) method is chosen when we have evidence that one classifier gives better performance than others. The overall performance of ensemble classifier can be improved by assigning higher weights to the decision values of that classifier. Experimental results show that the SVE method not only outperformed individual classifiers but also a simple majority voting ensemble method. A detailed discussion on the SVE model can be found in [46].

III. Methods and Materials

Machine learning (ML) is an automated and practical approach in the presence of a large amount of data set. It has been used in many classification applications when a labeled data set is available and got a reasonable accuracy [33]. We must adjust some parameters, depending on the classifiers used. For example, we preprocessed the raw pixel data and decided features set and learning parameters. After that, base classifiers analyze the extracted features to build classification

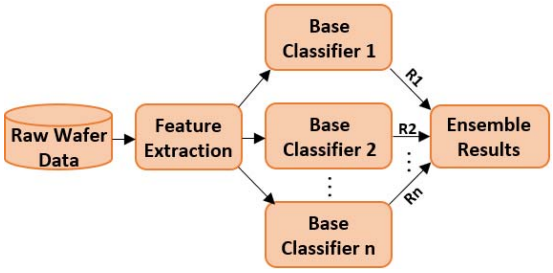

Fig. 1. A basic architecture of ensemble-based wafer map defect patterns identification.

models. All base classifiers have some implementation limitation, so the use of an ensemble approach (combination of base classifiers) is a common practice. Fig. 1 shows the basic steps of an ensemble-based WM classification approach. The first step was to prepare the raw wafer data by cleaning it from missing values. Then, useful multi-types features were extracted from input data, which were helpful to efficiently classify the WM defects. Proper features selection has the most impact on complexity reduction and accuracy improvement in any classification method [34]. Different ML classifiers were applied on the selected features and found the wafer defect recognition results for them. Lastly, these results were aggregated through the SVE approach to get final wafer defect classification result. The entire process is briefly described in the following subsections.

A. Data Collection

We used a real-life WM-811K dataset consisting of 811,457 wafer maps generated from 46,293 lots during circuit probe (CP) tests in a fabrication process. This is the largest open source WM dataset available at MIR lab website [35]. It also contains additional information about each WM such as die size, lot name, wafer index number, training/test labels, and failure types. Although each lot contains 25 WMs, so the total number of WMs should be 1,157,325 (i.e., 46,293 lots $\times$ 25 wafer/lot). However, not all lots contain exact 25 WMs because of sensor faults or some other unknown problem, thus they were removed. The die size (i.e., $l \times w$ image pixel size) of all WMs was different because each wafer image is two-dimensional and having different length and width pixel value. We found that there were 632 diverse sizes of WMs varying from (6, 21) to (300, 202).

B. Data Preparation

Our experimental data consists of 811,457 real WM entities, as shown in Fig. 2 Among them, we extracted the labeled dataset of 172,950 (21.3%) entities and excluded 638,507 (78.3%) entities with missing labels in attribute 'failure type', which can create a lot of problems in the knowledge discovery process of large dataset [36]. The labeled dataset also has two major classes such as pattern class and no-pattern class. The no-pattern class has no specific defect pattern of WM and labeled as *None*. It contains 147,431 wafer entities (85.2%) of the whole labeled data set. In addition, pattern class has only 25,519 wafer entities (14.8%) and consists of eight actual defect classes, labeled as *Center*, *Donut*, *Edge-local*, *Edge-ring*, *Local*, *Random*, *Scratch*, and *Near-full*. However, the data size for those classes is not equally distributed and consist of 4,294, 555, 5,189, 9,680, 3,593, 866, 1,193, and 149 WM entities respectively. The graphical representation of all labeled classes with selected samples (i.e., one random wafer from each class) is shown in Fig. 3. We applied random sampling to shuffle the data set and 5-fold cross-validation to train the classifiers which will be discussed in the next section.

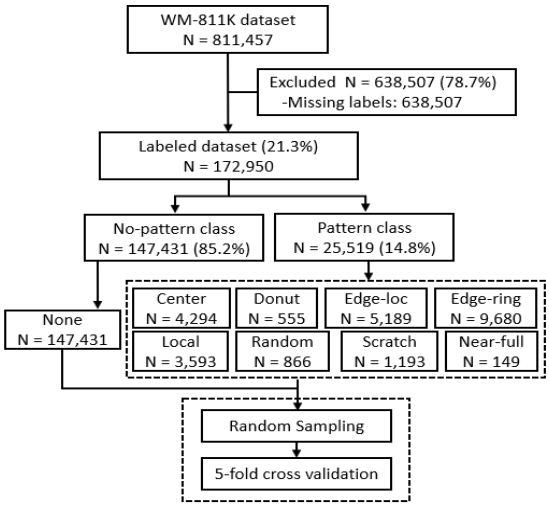

Fig. 2. Experimental data frequencies of different wafer map defect classes.

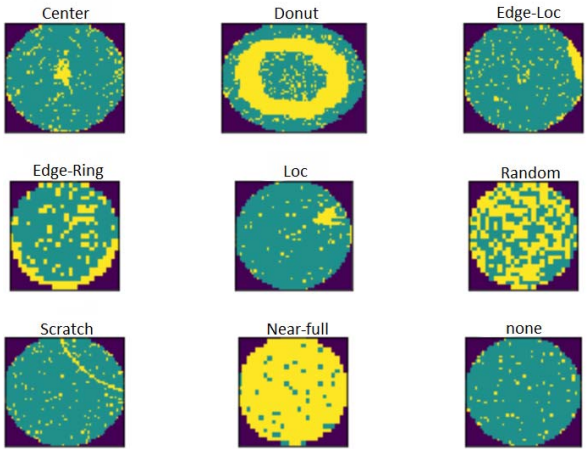

Fig. 3. Typical examples of WM defective patterns.

C. Features Extraction

Collection of useful features from raw data plays a vital role in WMDPI application. It can reduce the storage and computation cost by extracting hidden knowledge from actual data and provide discriminatory power in pattern recognition. We extracted multi-types features from each WM to make them ready for ML classifiers implementation, subsequent analysis, classification using scaling, attribute aggregation, and attribute decomposition. We proposed three distinctive features such as density-, geometry-, and radon-based. Density-based features depend on the failure density of various parts of WM image [15]. Geometry-based features are obtained from the geometric attributes of regions for each WM [47]. Radon-based feature is created by the transformation of radon, which

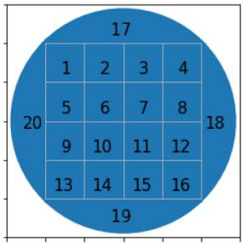

Fig. 4. Twenty parts distributions for density-based features.

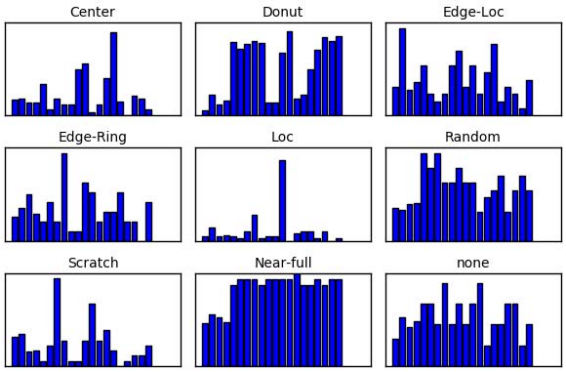

Fig. 5. Density-based features for selective samples.

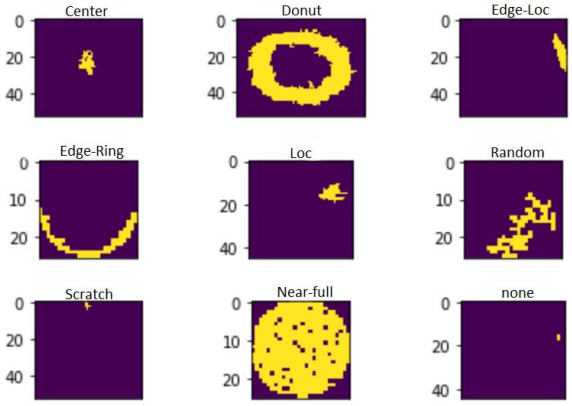

Fig. 6. Maximal area of most salient region for selective samples.

can create a 2-dimensional representation of WM based on a series of projections [37]. All these extracted features are manipulated together to show each WM with new representation. Detail description of all features is as follows.

1) Density-Based Features: For extraction of density-based features, each WM is divided into twenty parts and computed failure density of these parts. Out of 20, the initial 16 parts were similar and located in the center of WM. Remaining 4 parts were in the left, right, top, and bottom of WM, as shown in Fig. 4. We experienced that for different defect classes, defect density distribution for each part was also different. We can define it as follows:

i. *Center:* parts 6, 7, 10, and 11 contained high defect density.
ii. *Donut:* parts 1 to 16 excluding parts 6, 7, 10, and 11 contained high defect density.
iii. *Edge-Local:* one or more parts from 17 to 20 contained high defect density.
iv. *Edge-Ring:* parts 17 to 20 contained high and equal defect density.
v. *Local:* several parts from 1 to 16 contained high defect density.
vi. *Random:* each part contained almost similar defect density.
vii. *Scratch:* random defect density distribution.
viii. *Near-full:* each part contained almost 100% defect density distribution.
ix. *None:* no specific defect density distribution in any part.

Thus, we extracted twenty density-based features which represent corresponding parts of each WM. Fig. 5 shows these features for nine typical defect classes with selective samples. It turns out that density-based features are reasonable and makes the WMs more classifiable.

2) Geometry-Based Features: Geometry-based features define the geometric attributes of WM. The connected dice in WM creates regions that may cause of a specific defect pattern. The most salient region and its identification can be regarded as noise filtering. We used a region-labeling algorithm and selected the maximal area region as the most salient region which is proposed in a study [47]. Fig. 6 represents the most salient region with the max area for each WM defect class in selective samples. It also shows that only the actual defect patterns are visible and all other noises on each WM are removed. So, this feature also helps us to remove the random noise from the WM images. Based on max area salient region, we extracted six geometric features such as perimeter, area, length of minor axes, length of major axes, solidity, and eccentricity and detailed as follows.

i. *Perimeter:* the perimeter of the most salient region.
ii. *Area:* area of the most salient region.
iii. *Length of minor axes:* length of minor axes of the guessed eclipses surrounding the max area salient region.
iv. *Length of major axes:* length of major axes of the guessed eclipses surrounding the max area salient region.
v. *Solidity:* the proportion of defective dice in the guessed convex hull in the max area salient region.
vi. *Eccentricity:* the outline of the guessed eclipse surrounding the max area salient region.

3) Radon-Based Features: Radon-based features are created by the transformation of radon, which can create a 2-dimensional (2D) representation of WM based on a series of projections. A projection is designed by drawing several parallel rays from the 2D object of interest, conveying the integral of the contrast of object along with all rays to a single pixel in the projection. A collection of these projections from different angles is called a sinogram, which represents the original image into a linear transform. The radon transform results for nine typical defect classes with selective samples is shown in Fig. 7. However, even we got the radon transform results, we cannot extract features because all WMs are not of the same size. Thus, we used cubic interpolation to get fixed dimension features for row mean ($R\mu$) and row standard deviation ($R\sigma$). We extracted twenty features from each dimension, so we had forty radon-based features in total. Fig. 8 shows

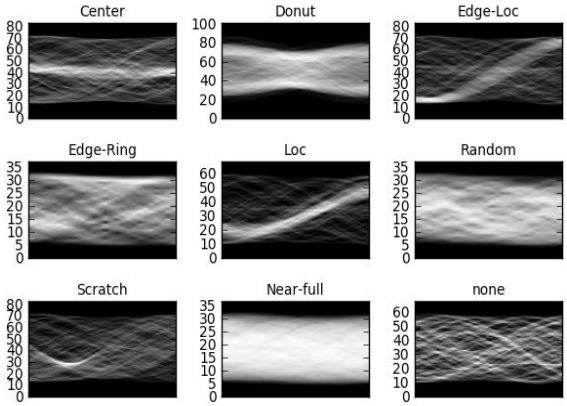

Fig. 7. Radon transform results for selective samples.

radon-based features of row mean and row standard deviation for each defective class in selective samples.

When we have extracted multi-type features such as twenty density-based features, six geometry-based features, and forty radon-based features for each WM, next step is to combine all these features together. As each individual feature represents a specific parameter of different wafer defects, their combination can play a vital role in defect identification. Thus, we have a total of sixty-six (i.e., $20+6+40 = 66$) effective features for each WM, which we used for ML analysis. Additionally, we applied feature extraction method on the actual wafer images and did not use any denoising technique, because denoising can destroy original defect patterns.

D. Machine Learning Classifiers

We used four state-of-the-art ML base classifiers such as logistic regression (LR), random forests (RF), gradient boosting machine (GBM), and artificial neural network (ANN) for WMDPI. However, the identification accuracy of individual base classifiers was different for different WM defect classes and no individual classifier got ideal accuracy. It was because each classifier has its own learning ability and parameter values. So, we used an ensemble approach which collected the best results of all classifiers and aggregated them to get the final classification result for all defect classes. The summary of all individual classifiers is given in the following paragraphs.

Logistic regression (LR) applies maximum-likelihood estimation, a commonly used method in ML algorithms, to find coefficients from training data [38]. It's a linear algorithm, so it creates a linear relationship between the input and output variables which built a more accurate model. A logistic function is used to describe the probabilities of possible outcomes for a single trial. Random forests (RF) also called decision tree forests is a meta estimator which focuses on combining various decision trees applied to different sub-samples of the original data set [28]. It combines basic principles of bagging by random selection of features for an additional diversity of the decision tree models. Finally, it uses the averaging to control over-fitting and improves predictive accuracy. It is because the size of all sub-samples is same as the input data set but drawn with replacement.

Gradient boosting machine (GBM) algorithm was designed for multiple classes of the logistic likelihood for classification, and Huber-M, least-squares, and least-absolute-deviation loss function for regression [39]. It produced highly robust and competitive methods for both classification and regression, especially for mining even less clean data. GBM builds a forward stage-wise additive model using gradient descent in function space. It sequentially constructs regression trees for all features in a completely distributed way. In each stage, nine regression trees (i.e., one tree for each wafer defect class) are fit on the negative gradient of the multinomial or binomial deviance loss function.

Artificial Neural Network (ANN or multilayer perceptron - MLP) defines the relationship between input signals and output signal using the same model as derived from the biological brain [40]. It contains at least three types of layers such as an input layer, one or more hidden layers, and an output layer and each layer contain various numbers of neurons or nodes. All nodes used activation functions to determine the result of the neural network like *Yes* or *No*. Activation functions are typically non-linear functions such as Sigmoid, Softmax, tanh, ReLU, and Leaky ReLU. The ANN implements a supervised learning approach called *Backpropagation* for training process. Its non-linear activation and multiple layers capabilities distinguish the ANN from a linear model. Formally, a one-hidden-layer ANN is a function $f : R^I \rightarrow R^O$, where I is the size input value x and O is the size of the output value $f(x)$, so in the matrix notation it can be defined as:

$$f(x) = A\Big(b^2 + w^2\Big(a\Big(b^1 + w^1 x\Big)\Big)\Big) \tag{1}$$

where b^1 and b^2 are the bias vectors, w^1 and w^2 are the weight matrices, and a and A are the activation functions.

E. Implementation Environment

Generally, raw wafer data contains noise due to complex fabrication process. Denoising of the data set can improve the defect identification performance for some specific classes, but it can destroy the actual failure patterns for other classes as well [4]. We used raw wafer data in our study which accompanied by noise so that, no defect pattern was destructed. However, some random noise was reduced by selecting the max area salient region during the feature extraction process. The dimensions of density-based, radon-based, and, geometry-based features were twenty, forty, and six, respectively. Thus, the overall feature dimension for each WM was sixty-six. For classification, an ensemble-based method was applied by joining the results of four well-known base classifiers such as LR, RF, GBM, and ANN. All statistical analysis was implemented using a personal computer with Windows10 operating system, Intel Core i7 CPU, and 16 GB RAM. It was developed using *scikit-learn* library [48] in an open source Web application of *Jupyter Notebook* which can be used for ML applications and Python language (version 3.5) [41].

F. Evaluation Methods

Normally a confusion matrix is used to evaluate the performance of a classifier. It is a specific table layout that represents the visualization of classification results. A normalized confusion matrix obtained by proposed SVE classifier is shown in Fig. 9. Where the numbers from 0 to 8 at

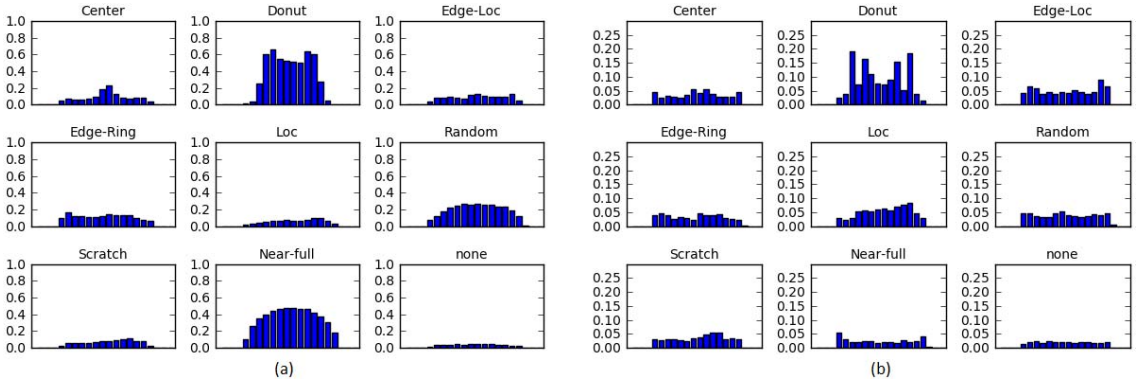

Fig. 8. Radon transformation results for each defective class in selective samples: (a) Radon-based features for row mean (Rμ) and (b) Radon-based features for row standard deviation (Rσ).

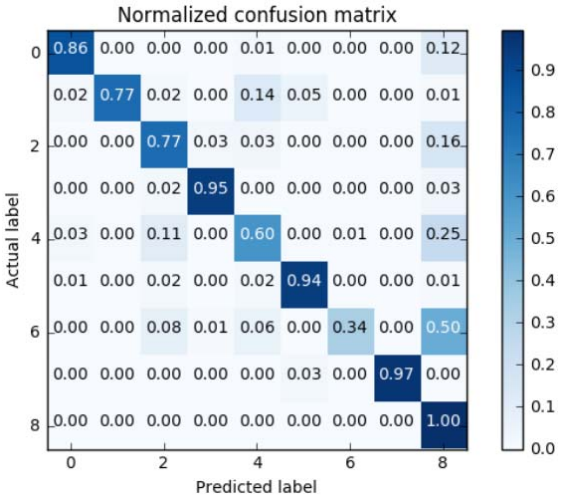

Fig. 9. A normalized confusion matrix of proposed SVE classifier.

x- and y-axis represent the labels of defect classes accordingly such as *Center*, *Donut*, *Edge-local*, *Edge-ring*, *Local*, *Random*, *Scratch*, *Near-full*, and *None*. Moreover, the *Actual label* in the y-axis represents actual class labels and *Predicted label* in the x-axis represents predicted class labels by the classifier. The diagonal values represent defect recognition rate of each class. This figure also shows that the proposed classifier got satisfactory defect recognition rate for all classes except *Local* and *Scratch* classes. The reasons for misclassification of these two classes will be discussed in the next section.

We built total five classifiers including one ensemble and four base classifiers and trained them with proposed multi-types features from whole data set. Prediction models of each classifier were generated through knowledge discovery from the training dataset. We selected five different types of classification performance measures, such as accuracy, precision, recall, F-measure, and AUC. Whereas, accuracy defines how often a classifier correctly predicts? The accuracy of a classifier can be obtained by equation (2).

$$Accuracy = \frac{All\ WM\ entities\ truly\ predicted}{All\ WM\ entities} \tag{2}$$

We selected precision and recall as two objective functions of each classifier and can be defined as:

$$Precision = \frac{WM\ entities\ truly\ identified}{All\ WM\ entities\ identified} \tag{3}$$

$$Recall = \frac{WM\ entities\ truly\ identified}{WM\ entities\ in\ standard\ test\ data} \tag{4}$$

Equations (3) and (4) show that while precision value increases the correctly tagged WM entities, recall value increases the tagged WM entities to the max. Moreover, there is an inverse relationship between precision and recall, and they represent two different classification measuring qualities. Whereas, F-measure is the weighted average (i.e., harmonic mean) of both precision and recall, and can be defined as:

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

Equation (5) shows that F-measure is the interpretation between predicted and actual results of WM defect entities. If these results are close to each other, the value of F-measure is high and vice versa.

Whereas, AUC is a performance measurement for all classifiers at different thresholds settings. It is a probability curve and shows the degree of separability of various defect classes. Higher the AUC value means better the classifier is at predicting each label class. By analogy, the higher the AUC value, the better the classifier is at distinguishing between all WM defect classes.

IV. Proposed Ensemble Approach

In this section, we discussed our proposed ensemble-based classifier which is developed by combining four different state-of-the-art base classifiers. Fig. 10 represents a flowchart of WM defect patterns identification with a classification ensemble approach. It consists of six basic parts such as data preparation, features extraction, training base classifiers, soft voting ensemble, defects recognition, and defects classification. First two parts are explained in the previous section. This section will explain the basic pillars of an ensemble approach. All ensemble approaches consist of three basic steps as diversity, training base classifiers, and joining the results of base classifiers. Each of these steps is further explained in the following subsections.

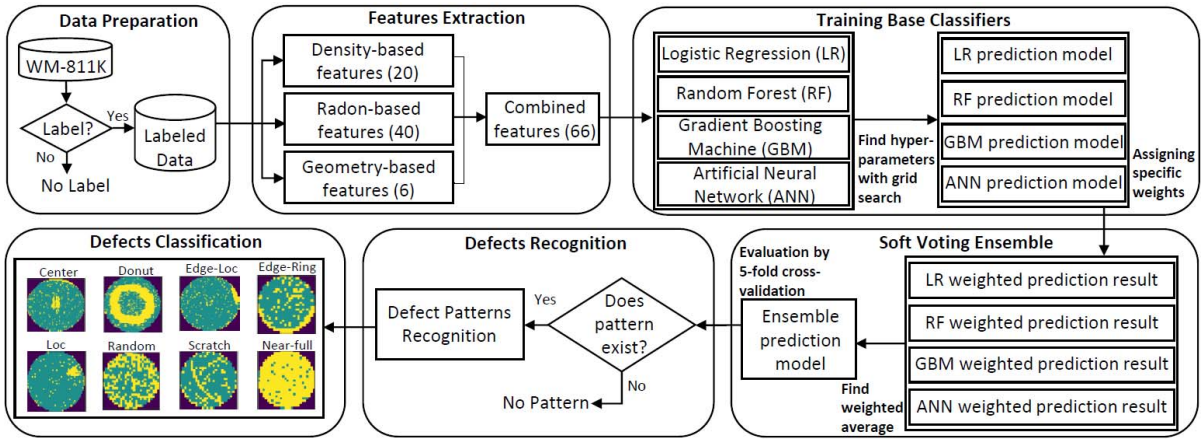

Fig. 10. A framework of soft voting ensemble classifier for wafer map defect patterns identification.

A. Diversity: Data Sampling

An ensemble model joins multiple classifiers and minimizes the error rate of individual classifiers. The error rate of individual classifiers varied with respect to their learning abilities. Diversity of an ensemble system defines that all base classifiers must be unique as much as possible and their decision boundaries should be different from each other. Use of different ML classifiers with various parameter boundaries is a popular method to improve the diversity of an ensemble classifier. All these classifiers are trained with same extracted features, so that they can generate various prediction models with same input data according to their decision boundaries. It makes sure that each classifier generated various prediction models under their corresponding training parameters.

B. Training Base Classifiers

There are many competing ML classifiers, but we used four of them to construct an ensemble model. Hyperparameters of each classifier were selected by grid search approach. For example, during training of the LR model, we use one-vs-rest (OvR) scheme in its *multi_class* parameter. It helps us to overcome the issue of multinomial classes. *L2* penalty was used for the regularization of the model by adding square of the magnitude of coefficients, where the values of coefficients are somewhere between 0 and ones for simple linear regression. *Liblinear* algorithm is used to handle optimization problems as it is the best option for OvR scheme [50]. The value of *n_estimator* was set to 100 for training of RF classifier, which means 100 decision trees were fitted in the forest. *Gini* criterion function was used to measure the quality of splits. Supported values of this function are *Entropy* for the information gain and *gini* for the Gini impurity [51]. The number of selected features is same as *square root* of *n_feature*. For the training of GBM classifier, 100 boosting stages were performed. This high value for boosting stages helps to minimize over-fitting issue which is very common during its training. *Deviance* loss function was implemented for its optimization which refers to classification through probabilistic outputs. The GBM classifier recovers the loss exponential by using AdaBoost algorithm. *Friedman_mse* function was used to measure the splitting quality which is a *mean square error* value of improvement score calculated by Friedman [52]. For the training ANN model, we used a single hidden layer with 100 nodes/neurons and applied the *ReLU* activation function, because of its effectiveness. It simply replaces the negative values with zero and positive values remain unchanged [53]. Moreover, the *ReLU* got better performance than other activation functions such as *tanh* and *Sigmoid*. The ANN model's training was terminated when validation score stopped improving. Moreover, *Adam* optimizer which is a stochastic gradient-based optimizer was used for weights optimization.

We applied a 5-fold cross-validation procedure for training the base classifiers. It randomly split the training dataset into 5 samples, and base classifiers use 4 samples for training purpose and remaining one sample is used to evaluate the models. The whole process is repeated 5 times so that each sample is used only once as the validation data. The five results are then aggregated to generate an ensemble prediction. Each classifier got various identification results for different wafer defects according to their learning rate, error rate, and other parameters. So, prediction results of all classifiers are combined with soft voting ensemble approach to obtain final prediction result which is more accurate than that of individual base classifiers.

C. Combining Base Classifiers

Final step of any ensemble method is to combine the results of individual base classifiers. There are many approaches to do this, but we preferred the most commonly used weighted average of soft voting ensemble (SVE) strategy. It is the combination of mean and weighted majority voting approaches. It applies weights directly to the continuous outputs instead of class labels. To decide weight values, we simply compare performance of all base classifiers and then assign higher weights to those classifiers with better results. Suppose that we have N weights, $w_1, \ldots, w_N$, generally obtained as performance of estimated generalization depending on training wafer data, with entire support for class w_1 as:

$$\mu_c(x) = \frac{1}{N} \sum_{n=1}^{N} w_n d_{n,c}(x) \tag{6}$$

After that, there are $N * C$ classes and specific classifier weights, which generates conscious class combination of classifier results. Finally, the entire support for class w_c is then:

$$\mu_c(x) = \frac{1}{N} \sum_{n=1}^{N} w_{n,c} d_{n,c}(x) \tag{7}$$

where $w_{n,c}$ is the weight of the n^{th} classifier for classification of class w_c. We trained the SVE model using same 5-fold cross validation as we discussed in previous section. Finally, by combining the results of four base classifiers an aggregated prediction result is generated.

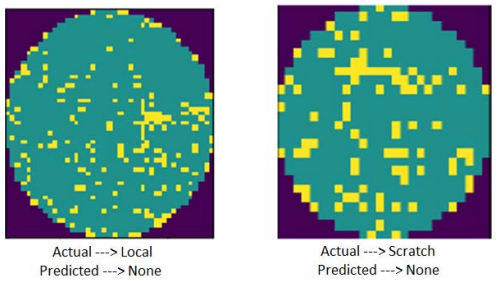

Fig. 11. Local and scratch wafer map defects are easily confused with none class due to prominent level of noise.

V. Evaluation and Discussion

This section provides a detailed overview of numerical experiments for recognition and classification of different WM defect patterns of the proposed ensemble approach.

A. Data Analysis

We used a real-world dataset WM-811K collected from 46,293 lots and consist of 811,457 original wafer images. Only 172,950 wafers were assigned with some defect patterns by domain experts. Eight defect classes and one normal class were defined. We extracted multi-types useful features such as twenty density-based, six geometry-based, and forty radon-based features from each WM. Set of these features play a vital role to identify various defect patterns from semiconductor wafers. So, we trained four state-of-the-art ML classifiers with same set of multi-types features but with different parameters and learning rates according to their nature.

B. Performance Evaluation

We implemented four different classification models namely, LR, RF, GBM, and ANN, using proposed multi-type features. Cross-validation with the 5-folds splitting process was used for training and validation of these models. Each model got various prediction results for different WM defect classes according to their learning abilities. Finally, we built an ensemble system by combining all four base classification models. We used the SVE technique to combine the results of the individual models. To do so, we trained all individual classifiers with same set of features extracted from original wafer image data set. Then we assigned higher weight values to the classifiers with higher prediction values and vice versa. Weighting the best classifiers more heavily increased the overall performance. We also used 5-fold cross validation for training and validation of SVE ensemble model.

Thus, we analyzed total of six WMDPI classification models, including four individual base classifiers and two ensemble classifiers such as majority-voting ensemble (MVE) and weighted averaging or soft-voting ensemble (SVE). Table I presents the performance measures such as precision, recall, F-measure, and AUC of these classifiers for all WM defect classes and best results are mentioned in bold numbers. It can be seen, that the SVE classifiers outperformed individual classifiers to identify all defect classes such as *Center*, *Donut*, *Edge-local*, *Edge-ring*, *Local*, *Random*, *Scratch*, *Near-full* and *None* classes with AUC values of 99.8068%, 99.9628%, 99.3354%, 99.9478%, 98.7545%, 99.9761%, 97.9009% 100.00%, and 99.5803%, respectively. The SVE classifier also got the highest value of F-measure score for most of defect classes such as *Center*, *Donut*, *Random*, and *Scratch*. Whereas RF got the highest value of F-measure score for *Edge-local* and *Edge-ring* defect classes, GBM for *Local* and *None* defect classes, and MVE for *Near-full* defect class.

Even though, the highest F-measure score for *Local* and *Scratch* classes was 69.7337% and 53.3333% respectively which is not a satisfactory result. The major reasons for this problem are unevenly distribution of wafer data set and prominent level of noise. Fig. 11 shows the examples of wafers from *Local* and *Scratch* classes which were misclassified to *None* class due to a high level of random noise. Although, we applied the max area on salient regions for denoising purpose, it resulted in no specific defect (i.e., *None or no pattern* class) on WM for these two classes.

Table II presents a comparison of different performance measures of the proposed SVE classifier with all base classifiers, MVE model, WMFPR method [4], and CNN classifier [44]. The architecture of proposed CNN classifier consists of an input layer, three sets of convolutional and pooling layers one after the other, a fully connected layer, and an output layer. The CNN classifier shows the overall values of accuracy, precision, recall, F-measure, and AUC for single defects as 89.8%, 93.6%, 95.6%, 94.6%, and 98.4%, respectively. In case of multiple defects (i.e., two or more), it shows accuracy, precision, recall, F-measure, and AUC values as 90.1%, 96.3%, 93.1%, 94.6%, and 98.6%, respectively. This CNN model was used to classify both single and multiple defect patterns, but we analyze only single defect classification result obtain with severely noisy data because our data set also contains random noise.

It can be seen from Table II that the proposed SVE classifier shows the highest performance with the overall accuracy, precision, recall, F-measure, and AUC values of 95.8680%, 96.8704%, 96.0570%, 96.8434%, and 99.9348%, respectively. Although the SVE classifier's results were not much improved than that of individual base classifiers, still it represents the power of the ensemble approach by getting the highest value for all performance measures. RF and GBM got second and third highest value with AUC score of 99.9021% and 99.8942%, respectively. It also shows that not only the proposed ensemble approach outperformed the WMFPR and CNN methods, but all individual base classifiers also got

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT CLASSIFICATION MODELS (%)

Classifier	Defect Type	Precision	Recall	F-measure	AUC
LR	Center	89.8802	77.8547	83.4363	99.4504
	Donut	81.7204	71.6981	76.3819	99.6567
	Edge-local	78.8340	58.6992	67.2796	97.2510
	Edge-ring	94.5220	92.1949	93.3439	99.7575
	Local	70.4918	31.0694	43.1294	93.0790
	Random	86.2500	77.5281	81.6568	99.9479
	Scratch	75.0000	16.1157	26.5306	91.2131
	Near-full	92.3077	85.7143	88.8889	99.9981
	None	96.4192	**99.8306**	98.0952	98.0474
RF	Center	**94.2181**	82.6990	88.0835	99.5730
	Donut	89.0110	76.4151	82.2335	99.4666
	Edge-local	**82.0896**	77.8302	**79.9031**	99.1632
	Edge-ring	**98.0645**	**95.5474**	**96.7896**	99.9310
	Local	**85.7451**	57.3699	68.7446	98.3578
	Random	95.1807	88.7640	91.8605	99.9735
	Scratch	**89.6552**	32.2314	47.4164	96.8953
	Near-full	96.5517	100.00	98.2456	99.9999
	None	97.7264	99.7797	98.7424	99.5038
GBM	Center	93.1336	86.0438	89.4484	99.6174
	Donut	88.4211	79.2453	83.5821	99.8419
	Edge-local	82.0697	75.5660	78.6837	99.2614
	Edge-ring	96.7327	94.6045	95.6568	99.8815
	Local	78.9762	**62.4277**	**69.7337**	98.5853
	Random	95.0617	86.5169	90.5882	99.9613
	Scratch	66.4474	**41.7355**	51.2690	95.4182
	Near-full	96.5517	100.00	98.2456	100.00
	None	**98.0909**	99.5967	**98.8381**	99.3694
ANN	Center	83.8470	**91.0035**	87.2788	99.7581
	Donut	84.2697	70.7547	76.9231	99.9486
	Edge-local	72.2807	77.7358	74.9091	98.9077
	Edge-ring	95.8289	93.8711	94.8399	99.8390
	Local	77.7480	41.9075	54.4601	97.7000
	Random	85.5422	79.7753	82.5581	99.8526
	Scratch	66.6667	38.8430	49.0862	95.2603
	Near-full	77.1429	96.4286	85.7143	99.9918
	None	98.1096	99.1968	98.6502	99.3087
MVE	Center	89.7862	87.1972	88.4728	N/A
	Donut	87.3684	78.3019	82.5871	N/A
	Edge-local	78.7512	**79.7170**	79.2311	N/A
	Edge-ring	97.5610	94.2902	95.8977	N/A
	Local	85.6459	51.7341	64.5045	N/A
	Random	**96.1534**	84.2697	89.8204	N/A
	Scratch	79.4118	33.4711	47.0930	N/A
	Near-full	**96.5517**	**100.00**	**98.2456**	N/A
	None	97.9520	99.6848	98.8108	N/A
SVE	Center	92.5428	87.3126	**89.8516**	**99.8068**
	Donut	**91.4894**	**81.1321**	**86.0000**	**99.9628**
	Edge-local	81.8002	78.0189	79.8648	**99.3354**
	Edge-ring	97.9415	94.7093	96.2983	**99.9478**
	Local	83.9130	55.7803	67.0139	**98.7545**
	Random	95.7831	**89.3258**	**92.4419**	**99.9761**
	Scratch	81.3559	39.6694	**53.3333**	**97.9009**
	Near-full	93.3333	100.00	96.5517	**100.00**
	None	97.9299	99.7187	98.8162	**99.5803**

*Note: LR denotes logistic reggression; RF random forest; GBM gradient boosting machine; ANN artificial neural network; MVE majority voting ensemble; SVE soft voting ensemble; and N/A not avialable

TABLE II
OVERALL EVALUATION RESULTS OF ALL CLASSIFIERS (%)

Classifier	Accuracy	Precision	Recall	F-measure	AUC
WMFPR	94.6300	N/A	N/A	N/A	N/A
CNN	89.8000	93.5750	95.6000	94.5767	98.4512
LR	95.0568	94.8425	95.4235	94.7624	99.7360
RF	94.4245	96.8410	96.9962	96.7445	99.9021
GBM	95.3488	96.7504	96.9673	96.8048	99.8942
ANN	95.2487	96.0833	96.2822	96.0360	99.8826
MVE	95.7442	96.7187	96.8951	96.6463	N/A
SVE	**95.8680**	**96.8704**	**96.0570**	**96.8434**	**99.9348**

*Note: WMFRP denotes Wafer Map Failure Pattern Recognition

TABLE III
COMPARISON OF COMPUTATIONALTIME

Method	Features set	Total time (s)	Avg. time (ms/wafer)
WMDPI	3	3655	21.6
WMFPR	2	12749	73.7

better results than these methods. Its major reason was the proposed set of multi-types features which were not given in previous studies. Most of the previous studies either used raw wafer data or only one or two features sets. However, we extracted three sets of features from wafer data which ultimately improved the overall performance of the proposed WMDPI method.

Moreover, most of the deep learning approaches including [42] and [44] applying for WM defect identification are using simulated data due to imbalance and noisy wafer data sets. In this way, they can improve the classification accuracy, but it becomes costly and time-consuming. Also, denoising the actual data can also destroy the actual defect patterns on the wafer images. But our proposed method got higher results even in the presence of imbalance and noisy data set.

We also compared all classifiers using *learning curves* which is a popular method to evaluate the performance of ML classifiers [49]. Learning curves show a measure of prediction performance of a classifier on a defined domain as a function of a measure of changing amount of learning effort. Fig. 12 presents the behavior of training score (red line) and cross-validation score (green line) of various classifiers. X-axis of each graph contains training data set and y-axis contains the accuracy score of each classifier. We can get an idea that how well different classifiers can generalize to new sample data. Learning curves of LR and RF show that both models are highly over-fitting because in RL's graph both training and cross-validation scores are plateaued and in RF's graph training score continuously remains at its maximum value (i.e., 100%) regardless of training sample variation. Whereas, training and cross-validation scores of GBM and SVE look similar and better than ANN's training and cross-validation scores. However, training and cross-validation scores of SVE are slightly better than that of GBM. So, SVE classifier has reduced the risk of over-fitting and under-fitting problems and reached the highest validation accuracy of 95.8680%.

Table III presents the comparison of computation time for features extraction between proposed wafer map defect pattern identification (WMDPI) method and WMFPR [4]. It is clear from the table that dispute of three diverse types of features

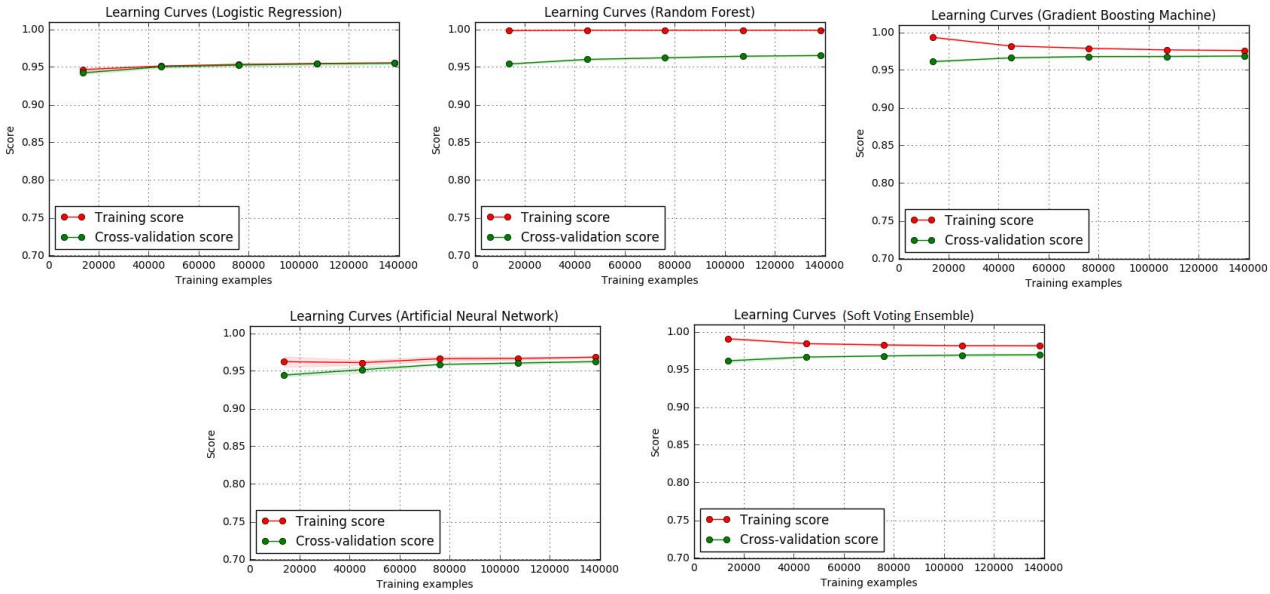

Fig. 12. Learning curves graphs of various machine learning classifiers.

sets, the WMDPI takes above three times less computational time as compared to the WMFPR, even in the presence of almost similar computational environments. It also makes our proposed method more feasible and efficient for wafer maps defects detection in a real-time fabrication process.

VI. Conclusion

In our study, we proposed an ensemble-based classifier with multi-types features to identify wafer map defect patterns and evaluated it for a real data set of WM-811K. We extracted three types of useful features: density-based, geometry-based, and radon-based, instead of using raw wafer images. These multi-types features provided a detailed description of original wafer maps and had improved the efficiency of the wafer defects identification system. An ensemble system was built by combining four state-of-the-art machine learning classifiers such as logistic regression, random forests, gradient boosting machine, and artificial neural networks. It joined the best prediction results of each classifier for different defect classes of wafer maps. The weighted averaging or soft voting ensemble approach was used to combine individual results and generate a final ensemble classification result. Experimental results showed that the proposed soft voting ensemble classifier outperformed all the individual classifiers and previously proposed methods with accuracy, precision, recall, F-measure and AUC score of 95.8616%, 96.7344%, 96.9326%, 96.7124%, and 99.9114%, respectively. It also reduced the problems of over-fitting and under-fitting while training and validation of model. Therefore, it turned out that soft voting ensemble approach was better than that of using single machine learning classifier in wafer maps defect patterns identification.

References

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

[2] C. H. Wang, W. Kuo, and H. Bensmial, "Detection and classification of defects patterns on semiconductor wafers," *IIE Trans.*, vol. 39, no. 12, pp. 1059–1069, 2006.

[3] J. Y. Hwang and W. Kuo, "Model-based clustering for integrated circuits yield enhancement," *Eur. J. Oper. Res.*, vol. 178, no. 1, pp. 143–153, 2007.

[4] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 1–12, Feb. 2015.

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

[6] A. Drozda-Freeman *et al.*, "The application and use of an automated spatial pattern recognition (SPR) system in the identification and solving of yield issues in semiconductor manufacturing," in *Proc. IEEE/SEMI Adv. Semicond. Manuf. Conf.*, 2007, pp. 302–305.

[7] J. Yu and X. Lu, "Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 1, pp. 33–43, Feb. 2016.

[8] C.-Y. Hsu, "Clustering ensemble for identifying defective wafer bin map in semiconductor manufacturing," *Math. Prob. Eng.*, vol. 2015, Jan. 2015, Art. no. 707358.

[9] C.-J. Huang, "Clustered defect detection of high quality chips using self-supervised multilayer perceptron," *Expert Syst. Appl.*, vol. 33, no. 4, pp. 996–1003, Nov. 2007.

[10] 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. 50, no. 12, pp. 3274–3287, Dec. 2012.

[11] C.-F. Chien, W.-C. Wang, and J.-C. Cheng, "Data mining for yield enhancement in semiconductor manufacturing and an empirical study," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 192–198, Jul. 2007.

[12] 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.

[13] M. Piao, C. H. Jin, J. Y. Lee, and J.-Y. Byun, "Decision tree ensemble-based wafer map failure pattern recognition based on radon transform based features," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 250–257, May 2018.

[14] J. Yu, "Fault detection using principal components-based Gaussian mixture model for semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 24, no. 3, pp. 432–444, Aug. 2011.

[15] M. Fan, Q. Wang, and B. V. D. Waal, "Wafer defect patterns recognition based on OPTICS and multi-label classification," in *Proc. IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. (IMCEC)*, 2016, pp. 912–915.

[16] P. Mohanaiah, P. Sathyanarayana, and L. G. Kumar, "Image texture feature extraction using GLCM approach," *Int. J. Sci. Res. Publ.*, vol. 3, no. 5, pp. 1–5, May 2013.
[17] S. Saha and A. Ekbal, "Combining multiple classifiers using vote-based classifier ensemble technique for named entity recognition," *Data Knowl. Eng.*, vol. 85, pp. 15–39, May 2013.
[18] P. N. Tan, M. Steinbach, and V. Kumar, "Methods for constructing an ensemble classifier," in *Introduction to Data Mining*, vol. 5. Boston, MA, USA: Pearson Educ., 2006.
[19] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer, 2012.
[20] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
[21] C.-F. Chien, S.-C. Hsu, and Y.-J. Chen, "A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence," *Int. J. Prod. Res.*, vol. 51, no. 8, pp. 2324–2338, Feb. 2013.
[22] C. W. Liu and C.-F. Chien, "An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing," *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1479–1486, 2013.
[23] 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.
[24] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, Sep. 2006.
[25] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[26] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[27] Y. Freund and R. E. Schapire, "Decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.
[28] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[29] J. A. Benediktsson and P. H. Swain, "Consensus theoretic classification methods," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 22, no. 4, pp. 688–704, Jul./Aug. 1992.
[30] L. I. Kuncheva, "Classifier ensembles for changing environments," in *Proc. Int. Workshop Multiple Classifier Syst.*, vol. 3077, 2004, pp. 1–15.
[31] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, no. 2, pp. 181–214, 1994.
[32] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognit.*, vol. 34, no. 2, pp. 299–314, 2001.
[33] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proc. Conf. Emerg. Artif. Intell. Appl. Comput. Eng. Real Word AI Syst. Appl. eHealth HCI Inf. Retrieval Pervasive Technol.* 2007, pp. 3–24.
[34] S. S. Gosavi, "Machine learning methods for fault classification," M.S. thesis, Inst. Comput. Archit. Comput. Eng., Univ. Stuttgart, Stuttgart, Germany, 2013.
[35] Mirlab.org. (2018). *MIR Corpora*. Accessed: Apr. 7, 2018. [Online]. Available: http://mirlab.org/dataSet/public/
[36] M. Saar-Tsechansky and F. Provost, "Handling missing values when applying classification models," *J. Mach. Learn. Res.*, vol. 8, pp. 1625–1657, Jan. 2007.
[37] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Harlow, U.K: Prentice-Hall, 2008.
[38] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. New York, NY, USA: Springer, 2013.
[39] J. Friedman, "Greedy boosting approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[40] R.-S. Guh, "On-line identification and quantification of mean shifts in bivariate processes using a neural network-based approach," *Qual. Rel. Eng. Int.*, vol. 23, no. 3, pp. 367–385, Mar. 2007.
[41] Jupyter.org. (2018). *Project Jupyter*. Accessed: Jun. 15, 2018. [Online]. Available: http://jupyter.org/
[42] T. Nakazawa and D. V. Kulkarni, "Wafer map defect pattern classification and image retrieval using convolutional neural network," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 309–314, May 2018.
[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Nueral Inf. Process. Syst.*, 2012, pp. 1097–1105.
[44] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 3, pp. 395–402, Aug. 2018.
[45] P. J. Boland, "Majority system and the Condorcet jury theorem," *Statistician*, vol. 38, no. 3, pp. 181–189, 1989.
[46] N. Littlestone and M. K. Warmuth, "Weighted majority algorithm," *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, 1994.
[47] R. M. Haralick and L. G. Shapiros, *Computer and Robot Vision*. Reading, MA, USA: Addison-Wesley, 1993.
[48] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 2, pp. 2825–2830, Oct. 2011.
[49] C. Perlich, *Learning Curves in Machine Learning*, IBM, Armonk, NY, USA, 2011. doi: 10.1007/978-0-387-30164-8_452.
[50] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
[51] B. H. Menze *et al.*, "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics*, vol. 10, p. 213, Jul. 2009.
[52] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[53] L. M. Andrew, Y. H. Awni, and Y. N. Andrew, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, pp. 1–6.