AdaBalGAN: An Improved Generative Adversarial Network With Imbalanced Learning for Wafer Defective Pattern Recognition

Junliang Wang, Zhengliang Yang, Jie Zhang, Qihua Zhang, and Wei-Ting Kary Chien

***Abstract*—Identification of the defective patterns of the wafer maps can provide insights for the quality control in the semiconductor wafer fabrication systems (SWFSs). In real SWFSs, the collected wafer maps are usually imbalanced from the defective types, which will result in misidentification. In this paper, a novel deep learning model called adaptive balancing generative adversarial network (AdaBalGAN) is proposed for the defective pattern recognition (DPR) of wafer maps with imbalanced data. In addition, a categorical generative adversarial network is improved to generate simulated wafer maps in high fidelity and classify the patterns with high accuracy for all defective categories. Taking consideration of the various learning abilities of the DPR model for different patterns into account, an adaptive generative controller is designed to balance the number of samples of each defective type according to the classification accuracy. The experiment results indicated that the proposed AdaBalGAN model outperforms conventional models with higher accuracy and stability for the DPR of wafer maps. Further results of comparative experiments revealed that the proposed adaptive generative mechanism can enhance and balance the recognition accuracy for all categories in the DPR of wafer maps.**

***Index Terms*—Semiconductor manufacturing, wafer defects, pattern recognition, generative adversarial networks.**

I. Introduction

THE SEMICONDUCTOR manufacturing process consists of four basic parts: wafer fabrication, wafer test, assembly and final test [1]. The integrated circuits are fabricated layer-by-layer in each die on a piece of silicon wafer [2], [3]. After the wafer fabrication, wafer test is conducted to evaluate the electrical function of each die [4]. Then, these qualified dies are cut from the wafer, assumed into individual chips and label as qualified product after the final test [5]. During the wafer testing, the detection results are saved on a wafer map [6], which is shown in Fig. 1. Analyzing the distribution pattern of defective dies on the wafer map is critical in the operation of the semiconductor wafer fabrication system (SWFS) [7], since the wafer map can provide insights to identify the reason of defects and improve the yield. With the development of semiconductor manufacturing technology, the design of integrated circuits becomes more complex and the number of the dies on a wafer is increased [8]. Moreover, the capacity of the SWFS is improved rapidly with the application of the automatic material handling system, robots, advanced planning system, etc. There is a great need for effective and efficient defect pattern recognition (DPR) in the operation of the SWFS [9].

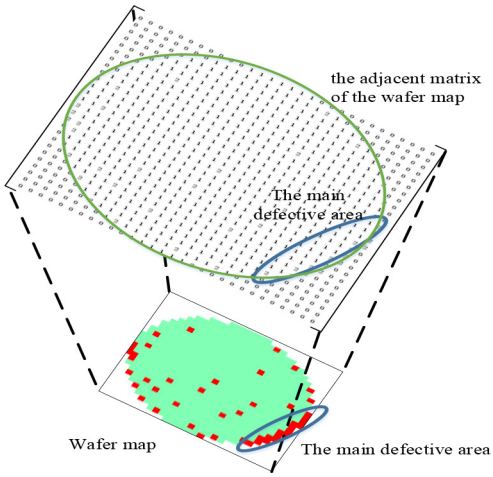


Fig. 1. The data structure of a wafer map.

Compared with conventional pattern recognition tasks, the DPR of wafer maps remains to be tough due to the imbalanced data in two sides:

1) The number of samples in different types is imbalanced, which means that one pattern in the DPR contains a much smaller (or larger) number of instances than the others. During the wafer manufacturing, the number of non-defective wafer maps (normal type) is usually much larger than the number of defective ones. Even within the defective maps, the number of records is various between different sub-classes of defective wafer maps. According to the wafer maps dataset collected in a real SWFS within two months, the defective wafer

Manuscript received May 5, 2019; revised June 15, 2019; accepted June 22, 2019. Date of current version August 2, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 51435009, in part by the Shanghai Sailing Program under Grant 19YF1401500, and in part by the Fundamental Research Funds for the Central Universities under Grant 2232019D3-34. *(Corresponding author: Jie Zhang.)*

J. Wang, Z. Yang, and J. Zhang are with the College of Mechanical Engineering, Donghua University, Shanghai 201620, China (e-mail: mezhangjie@dhu.edu.cn).

Q. Zhang is with the College of Mechanical Engineering, Donghua University, Shanghai 201620, China, and also with the Center of Quality and Reliability, Semiconductor Manufacturing International Corporation, Shanghai 201203, China.

W.-T. K. Chien is with the Center of Quality and Reliability, Semiconductor Manufacturing International Corporation, Shanghai 201203, China.

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.2019.2925361

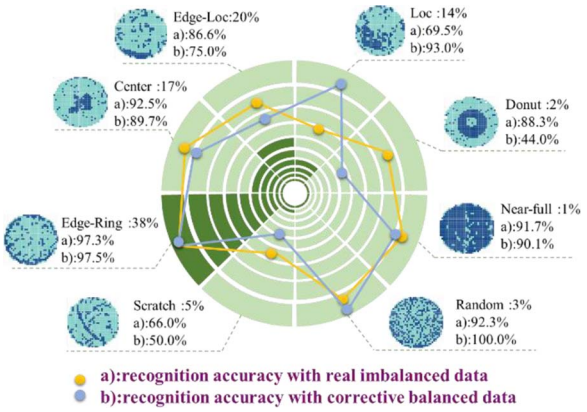


Fig. 2. The imbalanced data of an SWFS for the DPR of wafer maps.

maps can be divided into eight types, which is: loc, edge-loc, scratch, edge-ring, center, donut, random, and near-full. As shown in Fig. 2, the defective categories differ greatly in sample size. There are 38 percent wafer maps belong to the pattern named "edge-ring", whereas only one percent wafer maps are "near-full". Indicated by the previous studies [10], [11], the class imbalance is the main factor accounting for the bad performance in the pattern recognition, since they are biased toward the majority classes and tend to misclassify minority class examples [12].

2) The learning ability of the DPR method for different patterns is imbalanced. In the DPR, the majority and minority classes cannot be easily determined by the number of including instances, since the learning ability of the classifiers for pattern recognition is inequality between different classes [13]. One pattern recognizer may perform varying for the classification of different classes with the same sample size according to different data characteristics [14]. To verify this issue, two pre-experiments have been conducted with the same CNN model [15] for DPR of wafer maps. The results for real imbalanced dataset and artificial adjusted balanced dataset are shown in Fig. 2. As is shown in Fig. 2-b), the eight categories are adjusted with the CGAN model [16] by oversampling minority classes to have the same sample size, but receive different recognition accuracy. The recognition accuracy of the category named "scratch" is 50%, while the category named "random" is 100%. From another point of view, for some different datasets with high imbalanced rate, the minority class may be effectively recognized through many classification methods. For example, the category named "near-full" has only 1% number of instances in Fig. 2-a), nevertheless the recognition accuracy reaches up to 91.7%. Hence, the data imbalanced should be carefully treated during the DPR of wafer maps from two aspects: the imbalanced number of data instances and the imbalanced learning ability.

Motivated by the above mentioned issues, this paper proposed an adaptive balancing generative adversarial network (AdaBalGAN) for the DPR of wafer maps with imbalanced data. A conditional categorical generative adversarial network is designed to synthesize wafer maps in different kinds of defective patterns, which consists of a generator, a discriminator, and a classifier. Furthermore, considering the various learning ability of the recognizer to different defective patterns, an adaptive generative controller is proposed to modify the sample distribution of the wafer maps according to the accuracy of different defective patterns. To the best of our knowledge, it is the first work taking the imbalanced learning ability into account in the DPR of wafer maps.

The rest of this paper is organized as follows: Section II reviews the related studies about the DPR of wafer maps and imbalanced learning. Section III designs the adaptive balancing GAN with the categorical GAN model and the adaptive generative controller. Section IV reports numerical experiments of the defective pattern recognition based on wafer map datasets. Finally, conclusions and recommendations for future work are summarized in Section IV.

II. State of the Art

A. The Review of DPR of Wafer Maps

The DPR of wafer maps can be tackled as a pattern classification problem using machine learning technologies with the class labels annotated by human experts. The DPR has received considerable attention, and the proposed methods can be divided into three types: analytical methods, shallow structured artificial intelligent methods, and deep learning methods.

1) The analytical methods select the best matching defect pattern for each wafer map through a predefined defective function (such as: probability distribution function or spatial dependence function), which is designed during the data analysis. Taam and Hamada [17] designed a spatial dependence function to identify the spatial clustering type of the nonfunctional dies. Kim *et al.* [18] developed an infinite warped mixture model (essentially nonparametric Bayesian model) for the clustering of mixed-type defect patterns, which consists of several simple defective patterns.
2) The shallow structured artificial intelligent methods refer to the artificial intelligent methods (such as the RBFN, decision forest, and support vector machine [7], [19]), which can identify the pattern of the wafer defect based on the feature extracted by the pre-designed models. Wang [20] constructed a decision tree with the convexity and eigenvalue ratio to identify the specific defect type. Li and Huang [21] integrated the self-organizing map and support vector machine (SVM) for wafer bin map classification. Chao and Tong [22] presented a multi-class support vector machine with a defect cluster index for the DPR of wafer maps. Choi *et al.* [23] proposed a multi-step adaptive resonance theory (ART1) algorithm that includes variable resolution array and scaling strategy preprocessing to identify statistical models of four simulated defect patterns.
3) The deep learning methods recognize the defective patterns with deep neural networks, which can

extract features from raw data automatically [24], [25]. Nakazawa and Kulkarni [15] presented a convolutional neural networks (CNN) method for wafer map pattern classification and image retrieval, and the results indicated that the CNN model is effective in pattern recognition. Kyeong and Kim applied the CNN model to classify mixed-typed defect patterns of wafer bin map [26]. The analytical methods can hardly tackle the DPR with complicated defective features, since it is not easy to predetermine the detective functions during the continuous production. The shallow structured artificial intelligent methods are often applied as effective means to identify wafer map defects, however, they appear to be limiting if the input features are of low quality. Usually, the features of the wafer maps are extracted using techniques such as correlogram and Radon transform [27], which will result in loss or distortion of information to reduce the accuracy of pattern recognition [8]. The deep learning methods can extract the feature automatically, which is a promising method in the DPR of wafer lots. Especially, the CNN model can obtain surprising features even if the input data are incomplete or noisy, which is a part of the methodology proposed this paper.

B. The Review of Imbalanced Learning

Despite the existence of such data mining algorithms applied in DPR of wafer maps, there is no consensus about the imbalanced data to be taken into account [26]. In other pattern recognition problems, a number of approaches have been proposed to solve the classification tasks with imbalanced data, which can be categories into two types: model modifying method, and data processing method [12], [28]. The former treats the data instances from different classes distinctively, such as cost-sensitive learning [29], active learning [30], kernel-based method [31]. The data processing method (i.e., sampling strategies [32], and synthetic data generation [33]) changes the original data distribution by adjusting the number of instances to balance the data distribution, which is widely implemented and improved in the imbalanced learning. The downsampled strategies method reduces the data instances of the majority classes during imbalanced learning, which may potentially lose useful information during the instance reduction. Kang *et al.* [34] proposed a distance based weighted downsampled scheme for SVM for imbalanced classification. On the opposite, the oversampling method synthetic data instances to augment the minority classes to change the distribution of classes. Zhang *et al.* [35] designed a weighted minority oversampling (WMO) to balance the data distribution during fault diagnosis of rotating machinery. In the imbalanced learning of the DPR of wafer maps, the number of some minority class is quite small. For a typical semiconductor wafer fabrication system with the total capacity around 90,000 wafers per month, there may be only 900 pieces of wafer with a "near-full" wafer map. Under the circumstances, generating minority instances seems to be a feasible consideration.

Among the data generation methods, generative adversarial nets (GAN) was a hot topic in both academia and industry [36]. GAN can produce high simulation samples via an adversarial process, in which we simultaneously train two models: a generative model capturing the data distribution, and a discriminative model estimating the probability that a sample came from the training data rather than the generative model. Based on the GAN, a lot of various models have been designed to improve the generative performance, such as the InfoGAN [37], ACGAN [38]. Among these models, the conditional generative adversarial nets (CGAN) is proposed to generate data conditioned on class labels [16], [39]. With the CGAN, the minority class can be quantitatively oversampled to rebalance the wafer maps. However, the imbalanced learning ability of the classification methods for different samples have rarely be concerned by the current specialized instance generating methods. Actually, to determine the learning ability of a specific DPR method for different types of samples is a more fundamental issue to explore the nature relationship between the DPR method and the imbalanced data, which can contribute to the accuracy improvement of DPR.

III. The Adaptive Balancing GAN Approach

This section describes the framework of the proposed adaptive balancing GAN approach for imbalanced learning, which consists of two parts: the categorical generative adversarial networks and the adaptive generative controller.

During the wafer fabrications, the wafer map m_i for the i^{th} wafer is generated by the electrical property test machine, which is a matrix shown in Fig. 1. Each element x_{ij} in the wafer map m_i denotes if the electrical property of a die is qualified. The value of x_{ij} is allowed to be $\{0, 1, 2\}$ in this paper, where 0 means there is no die corresponding to the x_{ij}, 1 indicates the die corresponding to the x_{ij} is qualified, and 2 signifies that the corresponding die fails in the wafer test. The size of the wafer maps changes according to different layouts of the dies on the wafers, i.e., the size can be 45*48, 53*58, 26*26, 60*40. Hence, to standardize the input data, the toolkit of the cubic interpolation method in 'OpenCV Computer Vision' [40] is introduced to transform and resize the raw wafer map m_i, and the transformed wafer map is defined to be M_r.

A. The Categorical Generative Adversarial Networks for Data Generation

Aiming to re-balance the wafer map dataset, a conditional categorical generative adversarial network is proposed to generate the simulated instances for minority class, which has three parts: the pattern recognition model, the generative model, and the discrimination model. The first part is the pattern recognition model $f_c(.)$, which is a CNN model classifying the input wafer maps. The second part is a generative CNN model $f_g(.)$, which generates the synthetic wafer maps M_s according to the transformed real samples M_r. To guarantee the similarity between the M_s and M_r, a discriminator $f_d(.)$ is designed to measure the difference between the simulated wafer maps and real wafer maps. If the discriminator can recognize the M_s easily, the generator needs to improve the model performance. The objective functions of the generator

TABLE I
THE NETWORK STRUCTURE OF THE PROPOSED ADABALGAN

	Layer number	Layer	input	Output	Activation function	Batch_norm
The generator model $f_g(.)$	1	FC layer	[n,109]	[n,8*8*128]	LReLU	No
	2	Decon layer	[n,8,8,128]	[n,16,16,64]	LReLU	Yes
	3	Decon layer	[n,16,16,64]	[n,32,32,1]	Sigmoid	No
The discriminator model $f_d(.)$	1	Con layer	[n,32,32,1]	[n,16,16,64]	LReLU	No
	2	Con layer	[n,16,16,64]	[n,8,8,128]	LReLU	Yes
	3	FC layer	[n,8*8*128]	[n,1]	Sigmoid	No
the pattern recognition model $f_c(.)$	1	Con layer	[n,32,32,1]	[n,16,16,64]	LReLU	No
	2	Con layer	[n,16,16,64]	[n,8,8,128]	LReLU	Yes
	3	FC layer	[n,8*8*128]	[n,1024]	LReLU	No
	4	FC layer	[n,1024]	[n,9]	Softmax	No

and the discriminator are opposing and conflicting, and the performance of the two models are spiraling improved during the adversarial learning. In the model design of $f_d(.)$ and $f_c(.)$, the classical network Le-net5 [41] was referred to design the convolution kernel size in each layer, since the MNIST dataset has the same size of wafer maps (32*32 in this paper). Hence, the structure of $f_d(.)$ is improved with two convolution layers, and a fully connected layer. In the pattern recognition model $f_c(.)$, a Softmax layer is added as an output layer for classification. Aiming to generate adversarial simulation wafer maps in the AdaBalGAN, an opposite structure of $f_d(.)$ is presented with a fully connected layer, and two deconvolution layers. In the AdaBalGAN, the LReLU is selected as the activation function in input and hidden layers for better convergence. According to the recommendations given in DCGAN [42], the batch normalization is not selected in the input and output layer of $f_g(.)$ and $f_d(.)$.

The generator model $f_g(.)$ analyzes the real wafer maps and yields the simulated wafer maps with the input signal consisting of a noise signal and a conditional parameter, which can be formulated as: $M_s = f_g(\varepsilon, sign)$. In the $f_g(.)$ model, ε means the noisy signal, which is a vector of length 100, and each element is a random number from $[-1, 1]$. And the *sign* is a conditional instruction command directing the $f_g(.)$ to generate the corresponding wafer maps, which is a binary vector of length 9 (the one to eight bit corresponds to a defective category and the ninth bit corresponds to the normal wafer maps). For example, if the *sign* is [0, 1, 0, 0, 0, 0, 0, 0, 0], the $f_g(.)$ should output the wafer maps in the type of 2^{th} defective: edge-Loc. The network contains three layers of neurons, a fully connected layer (FC layer) and two deconvolution layers [42] (Decon layer) (the network structure is detailed in Table I). The output of the generator model is a matrix M_s with the same size as M_r.

The discriminator model $f_d(.)$ can be defined as $p_i = f_d(inp_i)$, which takes either a simulated wafer map or a real wafer map as input, and outputs the probability p_i to judge if the input instance is a real wafer map. Hence, the discriminator is a classification model with dual kinds of input data sources. There are three layers in the discriminator: two convolution layers (Con layer) and one FC layer. The input inp_i to the $f_d(.)$ is a simulated wafer map M_s or a real wafer map M_r. The input wafer maps are convoluted with small convolution kernels of size 4×4 pixels. The number of kernels increases from 64 in the first layers, to 128 in the second layer. And the third layer outputs a probability that describes whether the input is a synthetic wafer map.

The classifier model $f_c(.)$ recognizes the pattern of real wafer maps into different defective types, which can be defined as $Y = f_c(inp_i)$. The network structure with four layers of neurons is shown in Table I, which consists of two Con layers and two FC layers. The Con layers take the convolution kernels with the same size as the $f_d(.)$. And the number of kernels is 64 in the first layers, then increase to 128 in the second layer. The input of the $f_c(.)$ can be either a simulated wafer map or a real wafer map. The output Y is a probability vector of length 9, which is produced by the FC layer with the softmax function.

In the three networks of the AdaBalGAN, all trainable layers except the final layer use leaky rectified linear activation functions (LReLUs) to increase training stability [43]. The batch learning is taken in the training of AdaBalGAN, where the batch size is n in all three networks. For better model training, batch normalization [44] is used in the middle convolution layers, but not directly after the input layer or before the output layer.

The model training process consists of two phases. The generator, discriminator, and classifier are first trained to generate high-fidelity wafer maps of a specified category. In this stage, the switch in Fig. 3 is off, which indicates that the classifier $f_c(.)$ should learn to classify the real wafer map without the generated wafer maps, since the generated wafer maps are of poor quality. When the input samples are from real data, the cross entropy: $-\sum_{i=1}^{n}[y^i \log(f_c(M_r{}^i)) + (1-y^i)(\log(1-f_c(M_r{}^i)))]$ is minimized to improve the recognition accuracy, where y^i is a 0-1 value vector of length 9. There is only one "1" in y^i, whose index indicates the defective pattern number of the i^{th} wafer map. As a result, the objective function of the classifier

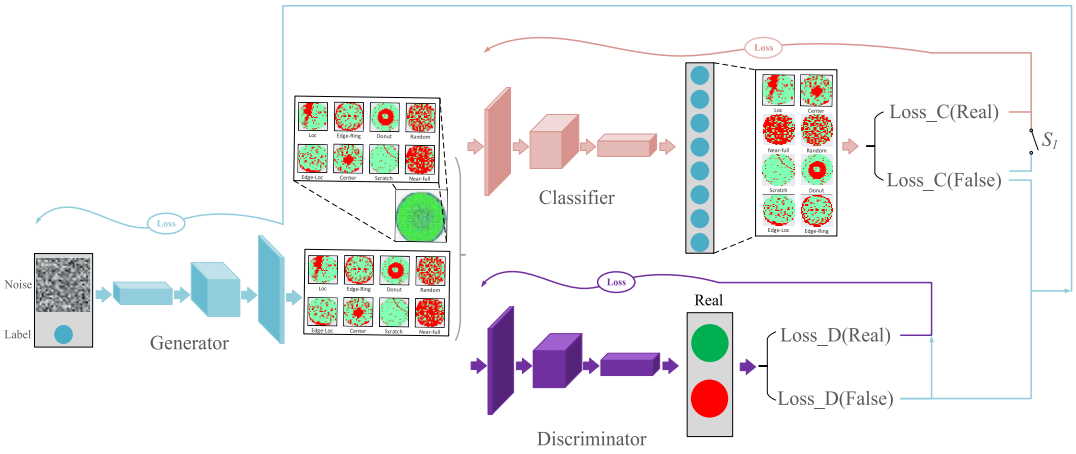


Fig. 3. The network structure of the proposed adaptive balancing GAN approach.

$f_c(.)$ during the model learning is defined as follows:

$$Obj_c = min_{f_c(.)} - \sum_{i=1}^{n}\left[y^i log\left(f_c\left(M_r{}^i\right)\right) + (1-y^i) \times \left(log\left(1-f_c\left(M_r{}^i\right)\right)\right)\right] \quad (1)$$

After the classifier was pre-trained, the discriminator $f_d(.)$ should learn to distinguish the simulated wafer maps and the real ones. When the input samples are created by the generator $f_g(.)$, the discriminator $f_d(.)$ should recognize the input instances to be the fake samples. As a result, the $\sum_{i=1}^{n} \log(f_d(M_s{}^i))$ should be minimized in the objection of the discriminator $f_d(.)$. On the contrary, the model $f_d(.)$ should accurately identify the collected wafer map instances from SWFSs as real samples, where the $\sum_{i=1}^{n} \log(f_d(M_r{}^i))$ should be maximized. Concretely, the objective function of the discriminator $f_d(.)$ during the model learning is formulated as follows:

$$Obj_d = min_{f_d(.)} - \sum_{i=1}^{n} log\left(f_d\left(M_r{}^i\right)\right) + \sum_{i=1}^{n} log\left(f_d\left(M_s{}^i\right)\right) \quad (2)$$

On the opposite, the generator $f_g(.)$ aims to maximize the probability of discriminator making a mistake and learn to create simulated data according to the *sign*. Therefore, the learning objection of the generator $f_g(.)$ should consist of two parts. The $\sum_{i=1}^{n} \log(f_d(M_s{}^i))$ is maximized to generate the wafer maps in order to confuse the discriminator. Moreover, the output of the classifier model is introduced to help the generator to produce the corresponding wafer maps according to the instruction command *sign*. Hence, the cross entropy: $-\sum_{i=1}^{n}[sign^i \cdot \log(f_c(f_g(\varepsilon^i, sign^i))) + (1 - sign^i)\log(1 - f_c(f_g(\varepsilon^i, sign^i)))]$ is minimized to make sure that the produced wafer map can be recognized as the corresponding pattern guided by the instruction command *sign*. The learning objection of the generator $f_g(.)$ is designed as follows:

$$Obj_g = min_{f_g(.)} - \sum_{i=1}^{n}\left[sign^i \cdot log\left(f_c\left(f_g\left(\varepsilon^i, sign^i\right)\right)\right) + \left(\mathbf{1} - sign^i\right)log\left(1-f_c\left(f_g\left(\varepsilon^i, sign^i\right)\right)\right)\right] - \sum_{i=1}^{n} log\left(f_d\left(M_s{}^i\right)\right) \quad (3)$$

At the second stage, only the classifier is trained to accurate identify the defective pattern for both real and synthetic wafer maps, since the generator $f_g(.)$ and the discriminator $f_d(.)$ has been trained. During the training, the switch S_1 in Fig. 3 is on, which means the classifier $f_c(.)$ should recognize the defective pattern of the input samples from both real and generated data. Therefore, the objective function of the classifier $f_c(.)$ during the model learning is defined as follows:

$$Obj_c = min_{f_c(.)} - \sum_{i=1}^{n}\left[y^i log\left(f_c\left(M_a{}^i\right)\right) + (1-y^i) \times \left(log\left(1-f_c\left(M_a{}^i\right)\right)\right)\right] \quad (4)$$

where $M_a{}^i$ refers to all wafer maps belong to the i^{th} category including real and generated wafer maps.

B. The Adaptive Generative Controller

During the imbalanced learning for the DPR of wafer maps, the learning ability of the pattern recognizer $f_c(.)$ for each class should be determined. During the pattern recognition, the learning ability is correlated with two types of factors: classification model and data property. The data property refers to the data distribution, the class composition, the overlapping between the classes, etc. The previous studies have shown that these data properties can seriously influence the recognition of the minority class [14]. However, to quantitatively analyze the relationship between these data properties and the learning ability is an arduous task, since the learning ability for a specific dataset is usually different from several specific learning models.

Hence, to measure the learning ability of the pattern recognizer to the wafer maps, this approach evaluates the classification performance of the pattern recognition model $f_c(.)$ with each class of wafer maps. The key idea of the proposed algorithm is to take the performance difference $accdif_k$ between the qualified wafer map pattern and the k^{th} defective wafer map pattern as a criterion. Different from the traditional methods evaluating the degree of imbalance by the number of instances, this method evaluates the recognition performance difference, which takes consideration of the imbalanced learning ability and the sample size. With the adaptive generative process (the

Algorithm 1: Accuracy recognition

Input: The set of counting variable t and $\Delta num_k(t)$,
the number of accurate classified samples for the k^{th} pattern $accnum_k$,
the sample size for the k^{th} pattern num_i,
the maximum training times t_{max}
the sample generating threshold h
the adaptive learning rate α
the momentum β
the standard sample size for a round G

Output: The recognition accuracy for each defectiv pattern of wafer maps acc_k

Initialize the adaptive learning ability, $t = 0$;
While $t < t_{max}$ or any $\Delta num_k(t) > h$ **do**
  $acc_k = \frac{accnum_k}{num_k}, acc_k \in [0,1]$;
  $acc_{max} = \max_k acc_k$
  $accdif_k = acc_{max} - acc_k$;
  $\Delta num_k(t+1) = \beta \times \Delta num_k(t) - (1-\beta) \times \alpha \times accdif_k \times G$;
  if $\Delta num_k(t+1) \geq h$ **then**
    invoking the GAN model to generate the simulation wafer maps and merge the generated instances into the corresponding class;
  end
  t = t + 1;
end

Fig. 4. The pseudocode of the adaptive generative controller.

pseudocode is shown in Fig. 4), the controller automatically decides the number of adjusted wafer maps for each defective pattern.

In the adaptive generative process, $\Delta num_k(t)$ refers to the number of generated wafer maps samples that the k^{th} defective pattern needs to increase at the t^{th} iteration. h means the sample generating threshold. If $\Delta num_k(t) > h$ and $t < t_{max}$, the oversampling process for the k^{th} defective pattern will be triggered. G is the standard sample size for each generation, working like a step-size factor.

The proposed adaptive generative controller works as follows:

- Initialize the controller, set counting variable $t = 0$, the number of generated wafer maps $\Delta num_k(0) = initial\ \Delta num$. *initial* Δnum is optimized by the pilot test.
- Calculate the recognition accuracy for each class of wafer maps with the classification method $f_c(.)$:

$$acc_k = \frac{accnum_k}{num_k}, \ acc_i \in [0, 1] \tag{5}$$

  where $accnum_k$ is the number of accurate classified samples for the k^{th} pattern, and num_k is the sample size for the k^{th} pattern.
- For each pattern, calculate the accuracy difference $accdif_k$ between the qualified wafer map pattern and the k^{th} defective wafer map pattern:

$$accdif_k = acc_{max} - acc_k \tag{6}$$

  where acc_{max} is the recognition accuracy for the qualified wafer map pattern (the majority class).
- For each pattern, the number of synthetic instances that needed to be generated is determined as follows:

$$\Delta num_k(t+1) = \beta \times \Delta num_k(t) - (1-\beta) \times \alpha \times accdif_k \times G \tag{7}$$

  Inspired by the error back propagation method [45], the adaptive learning rate α and momentum β are employed to accelerate the overall convergence of the accuracy difference.
- For each defective pattern, if $\Delta num_k(t+1) \geq h$, invokes the GAN model to generate the simulation wafer maps, then merges the generated instances into the corresponding class. h is the sample generating threshold.
- If $t > t_{max}$, or $\Delta num_k(t+1) > h$ is satisfied for every pattern, break the loop, otherwise, go back to the second step. t_{max} here is the maximum training times determined by pre-knowledge.

IV. Experiment Result

A. Experiment Setting for AdaBalGAN

The performance of the proposed GAN approach is evaluated with a dataset named "WM-811K" containing 811457 wafer maps, in which each wafer map was collected from real-world fabrication [27]. According to the imbalance rate defined by He and Garcia, the extended imbalance level is defined in the equation (8), where $\sum N_{maj}$ is the sum of the sample size of all majority classes, and the $\sum N_{min}$ is the sum of the sample size of all minority classes. Aiming to evaluate the performance under different imbalance levels, three sub-datasets were extracted from the "WM-811K" with low, medium and high imbalance levels (named as: D_l, D_m, D_h), where the imbalance rate for D_l, D_m, D_h are 1.56, 2.24, 2.80 in turn (shown in table II). For every dataset, the evaluation was performed by using 10-fold cross-validation. In each performance evaluation, datasets are divided into a training set, a pilot-testing set and a validation set in a ratio of 8:1:1. The proposed model is programmed with Python 2.7 [46] and Tensorflow1.0 [47].

$$r_{imb} = \sum N_{maj} / \sum N_{min} \tag{8}$$

During the model training at the first stage, the generator and discriminator are first learnt to synthesize wafer maps with the training set. Some real and generated wafer maps are showed and compared in Fig. 5. From the visual effect of the image, we can see that the proposed GAN model can generate high imitating wafer maps. The confidence of wafer maps in discriminator are shown in Fig. 6. The results indicate that the discriminator can identify the real and simulated wafer maps at first easily. But as the training continues, the confidence of real and simulated wafer maps are fluctuating around 50%, which means the GAN converges to generate high quality wafer maps.

After the generator and discriminator have been trained in the first stage, a pilot test is conducted to determine the initial Δnum (same for every category), h, and G. And the test results (shown in Fig. 7) indicate that the AdaBalGAN has the maximum accuracy when the initial Δnum equals 100, h is 10 and G is 500 for D_l. Furthermore, the initial Δnum equals 100, h is 0 and G is 750 for D_m. As for D_h, the indicated initial Δnum is 100, h equals 0 and G is 750. After the pilot test, the

TABLE II
THE THREE DATASETS WITH DIFFERENT IMBALANCED RATES

	Total	Loc	Edge-Loc	Center	Edge-Ring	Scratch	Random	Near-full	Donut	Normal
D_l	7570	1000	1000	1000	1000	1000	866	149	555	1000
D_m	12763	2000	2000	2000	2000	1193	866	149	555	2000
D_h	22356	3593	4000	4000	4000	1193	866	149	555	4000

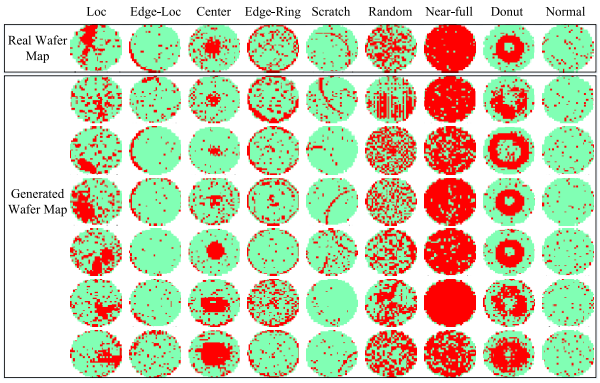


Fig. 5. The real and generated wafer maps.

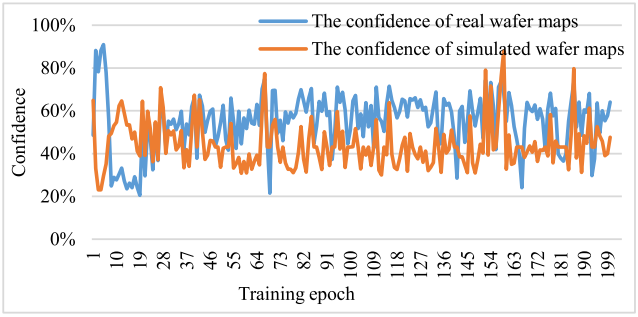


Fig. 6. The confidence of wafer maps in the discriminator.

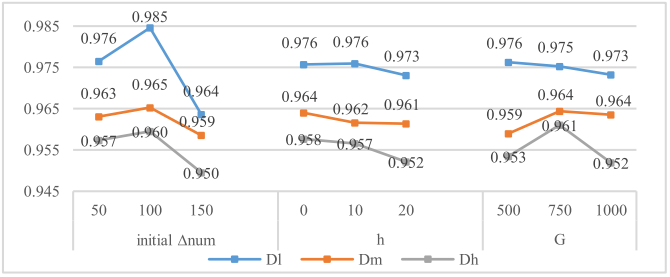


Fig. 7. The real and generated wafer maps.

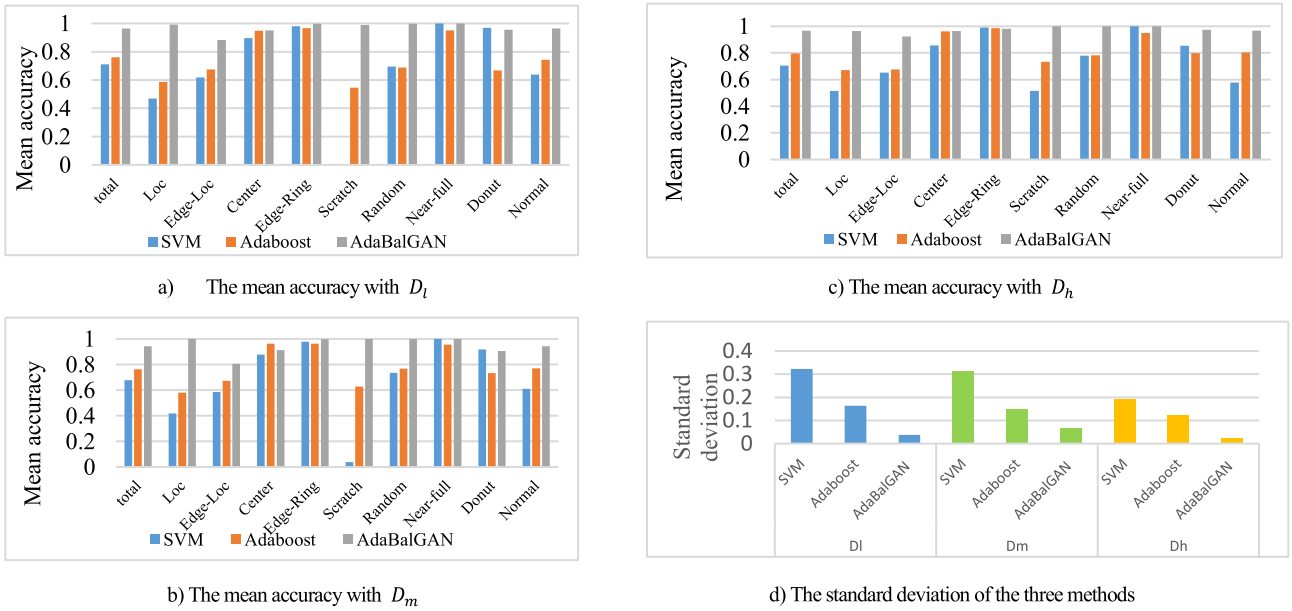
second stage of model training is performed for the classifier in the proposed AdaBalGAN.

B. Experiment for Defective Pattern Recognition

The performances of the AdaBalGAN were first compared with Adaboost [48] and SVM [49] with the validation set. The Adaboost is widely applied in the classification with imbalanced data, and the SVM can obtain probable results with the classification task even if the input data are incomplete or noisy. Adaboost uses 100 decision trees as weak classifiers, and SVM uses Gaussian kernels as kernel functions, both using recommended parameters in [48] and [49].

The generated wafer maps for different classes of three datasets are shown in Table III. And the experiment results about the DPR of wafer maps with three methods are shown in Fig. 8, which suggests that the proposed AdaBalGAN outperforms the SVM and Adaboost method in the DPR of wafer lots from the view of recognition accuracy. The advantage of the proposed AdaBalGAN is superior to the other two methods from the mean accuracy of the total wafer maps, the wafer maps of "loc", "edge-loc", "scratch", "random", and "normal". To evaluate the recognition performance for imbalanced data, the differences in the accuracy between all defective patterns are estimated. As shown in Fig. 8-d), the standard deviation of the accuracy between all defective patterns of the proposed AdaBalGAN is 0.038 with D_h, while the standard deviation is 0.32 of SVM and 0.16 of Adaboost. The results demonstrate that the proposed AdaBalGAN can both exactly and stably recognize the type of wafer maps, which is superior to the SVM and Adaboost.

The impressive performance of the proposed AdaBalGAN can be attributed to the generative adversarial mechanism and the adaptive generative controller. In order to illustrate the performance gains, the AdaBalGAN model was further compared with the CNN (same as the classifier in AdaBalGAN) and the GAN model (identical with the AdaBalGAN without the adaptive generative controller). As is shown in Fig. 9 a), b), and c), the GAN model outperforms the CNN model in the mean accuracy of the total dataset in D_l, D_m, and D_h, which means that the generative adversarial mechanism can improve the accuracy of DPR with imbalanced data. However, the improvement is limited for some minority categories, such as the wafer maps in the type of "loc" and "scratch". The GAN only takes consideration of the imbalanced number of samples, which creates the illusion of balanced data. The equalized sample size does not bring balanced recognition results. For example, the category named "near-full" is obviously a minority class from the number of samples. However, the CNN model can recognize this defective category with more than 80% accuracy. Moreover, the sample size of the category named "scratch" equals to the categories named "loc", "edge-loc", "center", and "edge-ring". Whereas, the CNN model performs badly in the DPR of the category named "Scratch", which is different from the other four categories. The results indicate that the various learning abilities of the DPR model for different patterns may lead to a significant difference in recognition accuracy.



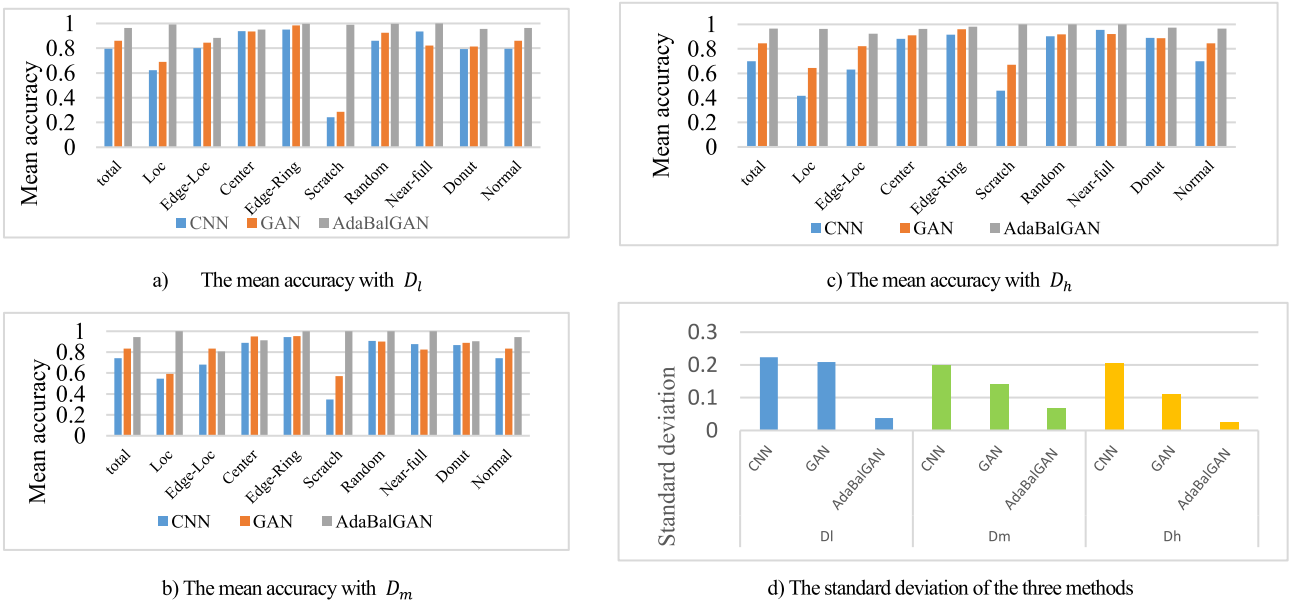
a) The mean accuracy with D_l

c) The mean accuracy with D_h

b) The mean accuracy with D_m

d) The standard deviation of the three methods

Fig. 8. The compared performance of AdaBalGAN, Adaboost, and SVM.



a) The mean accuracy with D_l

c) The mean accuracy with D_h

b) The mean accuracy with D_m

d) The standard deviation of the three methods

Fig. 9. The compared performance of AdaBalGAN, CNN, and GAN.

TABLE III
THE GENERATED WAFER MAPS FOR THREE DATASETS

	Loc	Edge-Loc	Center	Edge-Ring	Scratch	Random	Near-full	Donut	Normal
D_l	167	228	405	63	659	401	148	142	738
D_m	221	207	500	202	545	616	151	267	755
D_h	286	416	545	445	684	683	448	491	752

To further investigate the influence of the learning ability, the performance of the AdaBalGAN and GAN was compared and analyzed. As is shown in Fig. 9-d), the standard deviation of the AdaBalGAN is 0.038 with D_h, and 0.067 with D_m, and 0.025 with D_l, which is much smaller than the GAN model. The results indicate that the accuracy of the proposed AdaBalGAN is more stable than the GAN model in different patterns. As is shown in Fig. 9 a)-c), the proposed AdaBalGAN has close accuracy in most categories. For the dataset D_l, the accuracy of the proposed AdaBalGAN for the nine categories

TABLE IV
THE COMPUTING TIME OF THE TRAINING AND TESTING PROCESS

		Adaboost	SVM	CNN	GAN	AdaBalGAN
Time for training (s)	D_l	84.7	103.2	742.9	825.3	2187.5
	D_m	114.5	168.2	967.6	1284.6	3810.6
	D_h	186.3	284.6	1359.7	2364.3	6828.5
Time for testing (s/piece)		0.03	0.04	0.07	0.07	0.07

are close to 0.96. Especially for the categories named "scratch" (hard to be recognized), the accuracy of the AdaBalGAN reaches 0.98 with D_h, and both 100% right with D_m, and D_l. The results demonstrate that the AdaBalGAN with the generative adversarial mechanism and the adaptive generative controller can effectively recognize the defective patterns with imbalanced data.

C. Experiment for Computational Time

We have compared the five algorithms from the perspective of computational time, which is shown in Table IV. In the model training process, the total training time of the proposed AdaBalGAN is much higher than the Adaboost, SVM, CNN and GAN. This is because more training samples are taken in the adaptive generation of the simulated wafer maps. During the model evaluation, the computing time for the processing of a piece of wafer map is the same as the CNN and GAN, and a little bit higher than the Adaboost and SVM. Actually, the AdaBalGAN costs about seventy-two milliseconds for one piece of wafer map, which is quick enough for the DPR of wafer maps.

V. CONCLUSION

This study proposed an improved generative adversarial network with imbalanced learning for recognizing the defective patterns of the wafer maps produced by the electric property test, which can provide insights for the technical improvement of the SWFS especially during the productivity improvement phase. Compare with the conventional DPR work, this paper introduced imbalanced learning into the DPR of wafer maps and proposed a conditional generative adversarial network to synthesize simulated wafer maps. Furthermore, this paper designed an adaptive generative controller to adjust the generating rate according to the imbalanced sample size and learning ability of the DPR model. Results clearly show that AdaBalGAN model is much more accurate and stable than Adaboost, and SVM in the DPR of wafer maps. In addition, further experiments suggest that the different learning abilities of the DPR model to different sample classes should be taken into account in the DPR of wafer maps. The comparison among the proposed method, CNN and GAN model indicate that the adaptive generative mechanism can effectively significantly improve the recognition accuracy and stability.

Taking all points into consideration, future work will be paid on determining and measuring the learning abilities of a specific DPR model to the different defective patterns of wafer maps. In addition, the modelling and analysis of the mixed wafer defect, different imbalanced method (e.g., cost-sensitive learning method) will be considered to improve the wafer fabrication quality will be investigated in further research.

REFERENCES

[1] L. Münch, R. Uzsoy, and J. W. Fowler, "A survey of semiconductor supply chain models part I: Semiconductor supply chains, strategic network design, and supply chain simulation," *Int. J. Prod. Res.*, vol. 56, no. 13, pp. 4524–4545, Nov. 2017. doi: 10.1080/00207543.2017.1401233.

[2] J. Wang, J. Yang, J. Zhang, X. Wang, and W. Zhang, "Big data driven cycle time parallel prediction for production planning in wafer manufacturing," *Enterprise Inf. Syst.*, vol. 12, no. 6, pp. 714–732, Mar. 2018.

[3] J. Wang and J. Zhang, "Big data analytics for forecasting cycle time in semiconductor wafer fabrication system," *Int. J. Prod. Res.*, vol. 54, no. 23, pp. 7231–7244, Apr. 2016.

[4] K. B. Lee, S. Cheon, and C. O. Kim, "A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 2, pp. 135–142, Mar. 2017.

[5] W.-T. K. Chen and C. H.-J. Huang, "Practical 'building-in reliability' approaches for semiconductor manufacturing," *IEEE Trans. Rel.*, vol. 51, no. 4, pp. 469–481, Dec. 2002.

[6] S. F. Yang and W.-T. K. Chien, "Electromigration lifetime optimization by uniform designs and a new lifetime index," *IEEE Trans. Rel.*, vol. 64, no. 4, pp. 1158–1163, Dec. 2015.

[7] F. Adly *et al.*, "Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1267–1276, Dec. 2015.

[8] J. Wang, J. Zhang, and X. Wang, "A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 1, pp. 173–182, Feb. 2018.

[9] S. Kang, S. Cho, D. An, and J. Rim, "Using wafer map features to better predict die-level failures in final test," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 3, pp. 431–437, Aug. 2015.

[10] R. Razavi-Far, M. Farajzadeh-Zanjani, and M. Saif, "An integrated class-imbalanced learning scheme for diagnosing bearing defects in induction motors," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2758–2769, Dec. 2017.

[11] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on *K*-means and SMOTE," *Inf. Sci.*, vol. 465, pp. 1–20, Oct. 2018.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[13] B. Tang and H. He, "KernelADASYN: Kernel based adaptive synthetic data generation for imbalanced learning," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, Sep. 2015, pp. 664–671.

[14] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *J. Intell. Inf. Syst.*, vol. 46, no. 3, pp. 563–597, Jul. 2016.

[15] 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.

[16] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, Jan. 2018.

[17] W. Taam and M. Hamada, "Detecting spatial effects from factorial experiments: An application from integrated-circuit manufacturing," *Technometrics*, vol. 35, no. 2, pp. 149–160, May 1993.

[18] J. Kim, Y. Lee, and H. Kim, "Detection and clustering of mixed-type defect patterns in wafer bin maps," *IISE Trans.*, vol. 50, no. 2, pp. 99–111, Dec. 2018.

[19] 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.

[20] C.-H. Wang, "Recognition of semiconductor defect patterns using spatial filtering and spectral clustering," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1914–1923, Apr. 2008.

[21] T.-S. Li and C.-L. Huang, "Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 374–385, Jan. 2009.

[22] L.-C. Chao and L.-I. Tong, "Wafer defect pattern recognition by multi-class support vector machines by using a novel defect cluster index," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 10158–10167, Aug. 2009.

[23] 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, Jul. 2012.
[24] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.
[25] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 136–144, Jan. 2019.
[26] 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.
[27] 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.
[28] X. Gao, Z. Chen, S. Tang, Y. Zhang, and J. Li, "Adaptive weighted imbalance learning with application to abnormal activity recognition," *Neurocomputing*, vol. 173, pp. 1927–1935, Jan. 2016.
[29] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognit.*, vol. 40, no. 12, pp. 3358–3378, Dec. 2007.
[30] J. Hu, "Active learning for imbalance problem using L-GEM of RBFNN," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Xi'an, China, Nov. 2012, pp. 490–495.
[31] Y. Zhang, P. Fu, W. Liu, and G. Chen, "Imbalanced data classification based on scaling kernel-based support vector machine," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 927–935, Sep. 2014.
[32] S. Babu and N. R. Ananthanarayanan, "EMOTE: Enhanced minority oversampling technique," *J. Intell. Fuzzy Syst.*, vol. 33, no. 1, pp. 67–78, Jun. 2017.
[33] P. Lim, C. K. Goh, and K. C. Tan, "Evolutionary cluster-based synthetic oversampling ensemble (ECO-ensemble) for imbalance learning," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2850–2861, Sep. 2017.
[34] Q. Kang, L. Shi, M. Zhou, X. Wang, Q. Wu, and Z. Wei, "A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4152–4165, Sep. 2018.
[35] Y. Zhang, X. Li, L. Gao, L. Wang, and L. Wen, "Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning," *J. Manuf. Syst.*, vol. 48, pp. 34–50, Jul. 2018. doi: 10.1016/j.jmsy.2018.04.005.
[36] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Jun. 2014, pp. 2672–2680.
[37] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Jun. 2016, pp. 2172–2180.
[38] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2016, pp. 2642–2651.
[39] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computer Science*, p. arXiv:1411.1784, Nov. 2014.
[40] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *Proc. 35th Int. Conv. MIPRO*, Jul. 2012, pp. 1725–1730.
[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[42] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, Nov. 2015.
[43] S. H. Wang, P. Phillips, Y. Sui, B. Liu, M. Yang, and H. Cheng, "Classification of Alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling," *J. Med. Syst.*, vol. 42, no. 5, p. 85, Mar. 2018.
[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 37, Lille, France, Feb. 2015, pp. 448–456.
[45] J. Wang, J. Zhang, and X. Wang, "Bilateral LSTM: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 748–758, Feb. 2018.
[46] T. E. Oliphant, "Python for scientific computing," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 10–20, Jun. 2007.
[47] M. Abadi *et al.* (Mar. 2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. [Online]. Available: https://www.tensorflow.org/
[48] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost," *Stat. Interface*, vol. 2, pp. 349–360, Feb. 2006.
[49] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 1998.