IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 5, 2021, accepted March 20, 2021, date of publication March 24, 2021, date of current version April 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068378

Unsupervised Pre-Training of Imbalanced Data for Identification of Wafer Map Defect Patterns

HO SUN SHON[1], ERDENEBILEG BATBAATAR[2], WAN-SUP CHO[3], AND SEONG GON CHOI[4]

[1]Research Institute for Computer and Information Communication, Chungbuk National University, Cheongju 28644, Republic of Korea
[2]School of Electrical Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea
[3]Department of Management Information Systems, Chungbuk National University, Cheongju 28644, Republic of Korea
[4]School of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea

Corresponding author: Seong Gon Choi (choisg@cbnu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) by the Ministry of Education under Grant 2020R1A6A1A1204794511.

ABSTRACT Visual defect inspection and classification are significant steps of most manufacturing processes in the semiconductor and electronics industries. Known and unknown defects on wafer maps tend to cluster, and these spatial patterns provide valuable process information for supporting manufacturing in determining the root causes of abnormal processes. In previous studies, data augmentation-based deep learning (DL) techniques were most commonly used for the identification of wafer map defect patterns (WMDP). Data augmentation is an effective technique for improving the accuracy of modern image classifiers. However, current data augmentation implementations were manually designed for the WMDP problem. In this study, we propose a DL-based method with automatic data augmentation for the WMDP task. Basically, it focuses on learning effective discriminative features, from wafer maps, through a deep network structure. The network consists of a convolution-based variational autoencoder (CVAE) sequentially. First, we pre-trained the CVAE on large training data in an unsupervised manner. Second, we fine-tuned the encoder of the CVAE, which was followed by a neural network (NN) classifier, in a supervised manner. Additionally, we describe a simple procedure for automatically searching for improved data augmentation policies. The policy mainly consists of five image processing functions: rotation, flipping, shifting, shearing range, and zooming. The effectiveness of the proposed method was demonstrated through experimental results obtained from a simulation dataset and a real-world wafer map dataset (WM-811K). This study provides guidance for the application of deep learning in semiconductor manufacturing processes to improve product quality and yield.

INDEX TERMS Classification, convolutional variational autoencoder, deep learning, imbalanced data, neural network, unsupervised pre-training, variational autoencoder, wafer map defect patterns.

I. INTRODUCTION

In conjunction with the fourth industrial revolution, the semiconductor market has been expanding rapidly [1]–[3]. Semiconductor demand has been exploding in areas such as smartphones, virtual reality, automobiles, wearable devices, internet of things (IoT), and robotics [4]–[6]. Many diverse products are in demand. Semiconductor lines have become diverse, and the semiconductor fabrication process is complicated. Semiconductor manufacturers can produce semiconductor products with high yields and high quality to ensure market competitiveness. Semiconductor processes increase productivity through facility diagnosis, process control, stabilization of yield rate, and so on. In addition, the semiconductor fabrication process has been continually refined, and design complexity has increased to enhance productivity and semiconductor accumulation [7]–[9].

Semiconductor fabrication is conducted in two processes, from wafer fabrication to manufacturing the finished product. The first is the fabrication process of the integrated circuits on the wafer surface. The second is the testing process of the wafer map, processed by a unit die or chip after fabrication. As the fabrication process becomes more challenging and complicated, the number of defects increase. The processed

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

wafer was tested using the fabrication process, detailed later on, and subsequently assisted in identifying several defects [10]–[12]. As semiconductor manufacturing becomes complicated, and the difficulty of the refined process techniques increases, a new type of wafer defect map appears. This is because the generating mechanism according to the defect pattern of the wafer map is different. It is crucial to classify wafer maps automatically to eliminate the cause of defects.

Most of the steps used in semiconductor fabrication are conducted using a wafer map. If there are some abnormalities in the manufacturing process, defects will occur on the wafer. There are various types of defect patterns based on the manufacturing methods or features of abnormal unit processes. These defect patterns can be detected using wafer-map data from the test step of a wafer. To determine the abnormality process, causing wafer defects, at an early stage and to take steps to recover the yield rate, it is necessary to analyze the wafer map [13]. The process of sorting defective items among semiconductor fabrication processes involves electrical die sorting (EDS) [14]. It also tests the electrical motion state of each semiconductor chip generated on a wafer. To improve the yield rate of processing, engineers define and classify the forms of a defective wafer, and identify a wafer map, resulting in the EDS test [15]. Fig 1 shows an example of a wafer map. A large circle indicates a wafer, and small rectangles inside represent each die. The white color indicates that the die passed all the tests without any error, and other colors indicate that the die did not pass the test.

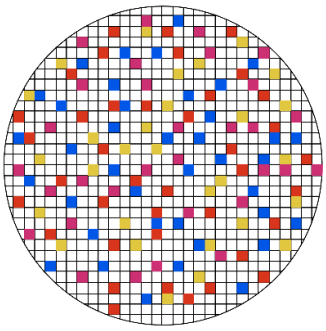

FIGURE 1. Example of a wafer map.

In ordinary semiconductor manufacturing companies, skilled experts classify and analyze defective patterns of wafer maps manually. However, when using this method, the classification performance of wafer map defect patterns can differ depending on the ability of the experts. Additionally, when production increases, it is difficult to cope utilizing only experts according to the growth of semiconductor demand [16]. Correspondingly, it is necessary to gain extra capacity to enable the system to cope during high productivity. The use of machine learning model, that learns the knowledge of experts, is one solution to increase capacity. Therefore, there is much research on handling these issues using machine learning or deep learning techniques. However, previous research faced some limitations. For example, there was a problem with classifying only defect patterns learning step. Also common problem in many data-oriented real-world semiconductor applications is class-imbalance [17]. Additionally, when the fabrication is refined and more complicated, the defect patterns of the wafer maps will vary. Therefore, it is necessary to develop a model that recognizes a new types of defective wafer map pattern.

Another common problem in many data-oriented real-world semiconductor applications is class-imbalance [17]. Previous studies have been conducted to classify wafer map defect patterns to handle imbalanced data and irrelevant features. In previous work, it was also handled by data augmentation [18]. Data augmentation is the process of extending the training data by applying class-preserving transformations such as rotation, flipping, shifting, shearing range, and zooming of an image to the original data. These processes have become important tools for achieving the accuracy of modern machine-learning algorithms. Data augmentation is a popular technique because of its simplicity in deep-learning applications. However, applying multiple transformations to the entire data set can increase the total size of the dataset tenfold, so data growth can be an expensive process. This may have some advantages in terms of overfitting, but it increases the overall training data. It can also significantly increase the cost of data storage and training time and can scale linearly or super linearly with respect to the training set size. Semiconductor engineers have applied many different techniques, such as manual visual inspection and machine learning algorithms, using manually extracted features for wafer defect classification. Therefore, automatic wafer map identification systems need to be developed by taking advantage of machine learning and deep learning methods.

In this study, we consider the data imbalance problem by developing a deep learning-based method. It automatically classifies wafer map defect patterns without manual data augmentation or feature extraction. We employed a convolutional neural network (CNN) to extract visual features from the wafer map images. A generative variational autoencoder (VAE) was used to learn the data distribution and sample augmented data. The data augmentation function includes transformations such as rotation, flipping, shifting, shearing range, and zooming. First, we pre-trained the convolutional variational autoencoder to learn training samples and generate augmented data. Then, we fine-tuned only the encoder part, followed by the neural network (NN) classifier for the classification of wafer map defect patterns.

The contributions of this paper are summarized as follows:

1. We proposed an automatic classification method that employs deep learning techniques, such as CNN and VAE, for wafer map defect patterns without manual data augmentation and feature engineering.
2. We designed a convolutional variational autoencoder (CVAE) that learns the distributions of visual data. Then, it also samples various data transformations to solve data imbalance problems.
3. We automated the process of finding an effective data-augmentation policy for a wafer map dataset. Each

policy expresses several choices and orders of possible augmentation functions, such as rotation, flipping, shifting, shearing range, and zooming.

4. Comprehensive experiments demonstrate that the proposed method can obtain good results for identifying wafer map defect patterns. By combining convolutional operations and a generative model, we can obtain competitive results with other state-of-the-art deep learning methods. Additionally, we generated wafer map images with various transformations for each non-defect and defect class.

The remainder of this paper is organized as follows. We first review related works in Section II. In Section III, we introduce the proposed method in detail. Section IV reports the experimental settings and results and provides a discussion and analysis. Finally, conclusions and future work are provided in Section V.

II. RELATED WORKS

Research has been conducted to classify defective wafers into each pattern using wafer map information. In this section, we review some recently published research that uses machine learning and deep learning.

In the early stages, research has been conducted to extract features from wafer maps and classify defective patterns using machine learning techniques. Machine learning classification algorithms classify the defective patterns based on the pre-defined visual features from the wafer map. Manually or automatically obtained features, using feature extraction techniques, have been explored in computer vision [19]. For example, the features were extracted from the wafer map using Hough transformation, and the defect ratio at the center of the wafer was calculated. A variety of machine learning classification algorithms then apply the extracted features to classify defect patterns [20], [21]. In addition, wafer map transformation into spatial correlation and dynamic time warping [22] techniques were used in feature extraction, and the defective patterns in the results were classified using the k-nearest neighbor classifier [23]. As studied in domain analysis, principal component analysis (PCA) introduces the pattern index of the wafer map, which produces indices and variables focused on the structural features of the wafer map [24]. Therefore, in the case of a wide fluctuation of produced items, there would be problems in conducting modeling again. Additionally, singular value decomposition was used to transform the wafer map into a regularized singular value, followed by a k-nearest neighbor classifier [25]. However, there are some problems in real-world situations when using k-nearest neighbors, which makes it difficult to achieve good performance when insufficient training data and high computation time are required. After projecting the wafer map into Radon the data is transformed into four feature subsets, namely, max, minimum, means, and standard deviation, an ensemble model was proposed based on the decision tree of each feature subset [26]. Moreover, applying density-based and geometry-based Radon techniques to ensemble classifiers, constructed with logistic regression, random forest, gradient boosting machine, and artificial neural net [27] suggested a model using the extracted features of the wafer map.

Recently, various techniques have been proposed for the identification of wafer map defect patterns by taking advantage of deep learning. For example, without feature extraction of wafer maps or spatial filtering, research has been conducted widely using CNN, which applies intact original images. In CNNs, the features necessary for classification learn for themselves through convolution layers [18]. In another recent study using CNN, a wafer map was constructed according to 22 defective patterns, defined in advance, and then using the map, the patterns were classified into convolutional neural networks and applied for image retrieval. Even though the classification model showed an accuracy of 98% for the artificial data, some patterns extracted from the real data showed an accuracy of 68%. This demonstrates the limitations of artificial data [28].

Moreover, Kyeong and Kim [29] proposed a CNN-based classification model to classify mixed-type defect patterns in wafer bin maps separately for each pattern circle, ring, scratch, and zone. Cheon *et al.* [30] proposed an automatic defect classification method based on deep learning that was designed to achieve high classification performance for known defect classes and also classify unknown defects. Jin *et al.* [31] proposed a clustering-based defect pattern detection and classification framework, based on the density-based spatial clustering of applications with noise. Ishida *et al.* [32] proposed a deep learning-based failure pattern recognition framework that only uses data augmentation techniques with noise reduction, without accessing a large amount of training data. Shen and Yu [33] integrated wafer map defect recognition with deep transfer learning, which reduces the training time and improves the feature learning performance. It also addresses the problem of class imbalance. Wang and Chen [34] used extracted features based on three types of masks: polar masks, line masks, and arc masks. These masks extract rotation-invariant features for classifying defect patterns. Yu [35] proposed an enhanced stacked denoising autoencoder with manifold regularization techniques to generate discriminative features from wafer maps. Yuan-Fu [36] used automatic optical inspection to visualize defect patterns and identify the root causes of die failures. Then, CNN and extreme gradient boosting methods are employed for wafer map retrieval and defect pattern classification. Shawon *et al.* [37] also modified the CNN architecture to improve the classification performance and used data augmentation techniques to solve the data imbalance problem. Nakazawa and Kulkarni [38] proposed a deep convolutional encoder-decoder neural network architecture for detecting wafer map defect patterns, as well as segmentation. Yu *et al.* [39] proposed a stacked convolutional sparse denoising auto-encoder for wafer map pattern recognition and a feature learning method to learn discriminative features from wafer maps. Yu and Liu [40] proposed a deep neural network, which is a two-dimensional PCA-based

convolutional auto-encoder for wafer map defect recognition. Alawieh *et al.* [41] used a deep selective learning technique and featured an integrated reject option where the model chooses to abstain from predicting a class label when the misclassification risk is high. Thus, there is a trade-off between the prediction coverage and the risk of misclassification. Jang *et al.* [42] proposed an ensemble model of a one-versus-one method that uses a CNN as the base classifier for wafer map classification, and then examined the open set recognition problem, in which wafer maps must be classified using major defect patterns. Tsai and Lee [43] proposed a CNN encoder-decoder-based data augmentation and depth-wise separable convolution-based defect classification. They also developed a classifier with a reduced-weight architecture based on depth-wise separable convolutions [44]. Yu *et al.* [45] addressed the problem of insufficient labeled images with various defects. They proposed a semi-supervised deep-learning-based transfer learning algorithm by joining features and labels in an adversarial network. Jin *et al.* [46] presented an image-based classification method for wafer map defect patterns without any specific preprocessing. They extracted high-level features from a CNN fed to a combination of error-correcting output codes and support vector machines for the classification of wafer map defect patterns. Wang and Chen [47] used polar mapping before training the CNN. Then, the circular wafer map was transformed into a matrix. They also applied a data augmentation technique to eliminate the effects of rotation. Saqlain *et al.* [47] addressed the data imbalance and irrelevant features problem using data augmentation techniques such as rotation, flipping, shifting, shearing range, and zooming of an image to the original data.

Owing to the limitations of previous studies, we developed a novel classification technique by modifying the CVAE. The modified CVAE automatically performs data augmentation without manual rules or large data generation. In addition, pseudo-data are generated from the distribution of each class label. The experimental results demonstrate the efficiency of the proposed method.

III. PROPOSED METHOD

In this section, we discuss the basic structure of the proposed method in detail. We also provide the training procedure and hyperparameter settings.

A. ARCHITECTURE

Wafer maps provide important information when represented as images for engineers to identify the root causes of die failures during semiconductor manufacturing processes. In computer vision, CNN is a deep learning-based technique commonly applied to analyzing visual imagery. In real-world problems, data imbalance is a critical issue. As we discussed, CNN is the basic technique adopted in the identification tasks of wafer map defect patterns, and data augmentation techniques are generally used for data imbalance problems. In this study, we employed CNN as our base feature learner. Instead of using manual data augmentation, generative models generate samples for high-dimensional datasets, learns the data distribution, and generates new samples from the learned distribution. We designed a CVAE that is improvised with image operations such as rotation, flipping, shifting, shearing range, and zooming for more effective image generation. We then used the basic NN technique for the classification of defect patterns. It calculates the probability distribution for each class label, and the maximum value is chosen for the final prediction. First, we pre-train the CVAE model by minimizing the reconstruction loss, and the mean square error was also used. Second, we train the NN classifier by minimizing cross-entropy loss. An overview of the proposed method is presented in Fig 2. As shown, we input wafer map images to the proposed method and identify whether they are defective or not. The common defect patterns are edge ring, edge local, center, local, scratch, random, donut, and near-full. In the following sections, we explain the proposed method in detail.

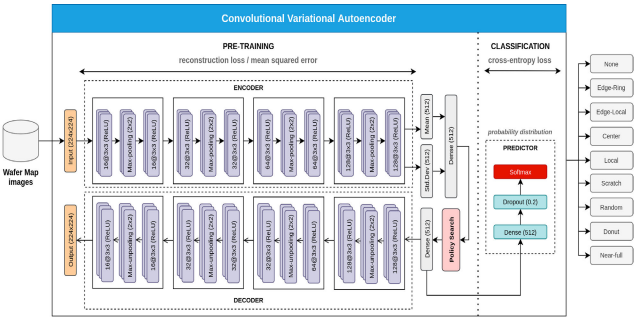

FIGURE 2. Overview of the proposed method.

1) CONVOLUTIONAL NEURAL NETWORK

A CNN is a type of deep neural network with the capability of extracting useful features by utilizing several convolutional operators. It is particularly suitable for two-dimensional data structures; therefore, it is a popular pattern recognition classifier in image processing.

In a CNN, as a weighted kernel K slides over every position of input data x, the convolution operation of the input data and kernel is triggered, resulting in a feature map:

$$S(i,j) = (X * K)(i,j) \tag{1}$$

$$= \sum_m \sum_n I(i-m, j-n) W(m,n) \tag{2}$$

where S is the feature map resulting from input data x and kernel K, and $*$ denotes the convolution operation.

Typically, the kernel size is smaller than the input data size, but with greater depth. This means that several different kernels are applied to the input data at the same time, resulting in the same number of feature maps. The weights of the kernels were adjusted during the training.

Although CNNs are mostly applied for the identification of wafer map defect patterns, they have also been successfully explored in fault classification and diagnosis in semiconductor manufacturing processes [48]. Because wafer map defect patterns have the same 2-dimensional data structures as images, the CNN for analyzing images is suitable for identification.

2) VARIATIONAL AUTOENCODER

VAE, an important generative model, has a similar network frame as an autoencoder, which consists of two parts: an encoder and a decoder. In the autoencoder, the encoder defines a mapping from input data $x \in \mathbb{R}^{d_x}$ to a latent variable $z \in \mathbb{R}^{d_z}$, while the decoder defines a mapping back from the latent variable z to the input space, which outputs the reconstructed $\hat{x}$. The training objective of the autoencoder is to make the reconstructed term $\hat{x}$ as close as the original one x, forcing autoencoders to learn the latent features of normal data. In VAE, the latent variable z is constrained to be distributed according to a prior distribution $p_\theta(z)$, usually a multivariate unit Gaussian $N(0, I)$, forcing the model to learn the distribution of input data. However, when mapping from the input data x to the latent variable z, according to Equation (3), $p_\theta(z|x)$ is usually intractable because $p_\theta(x)$ is also intractable.

$$p_\theta(z \mid x) = \frac{p_\theta(x, z)}{p_\theta(x)} \tag{3}$$

Hence, variational inference techniques are used to solve this problem in a tractable manner by finding an approximation posterior $q_\phi(z|x)$.

$$q_\phi(z \mid x) = N(\mu_z, \sigma_z^2 I) \tag{4}$$

where the mean μ_z and standard deviation q_z of the approximation posterior $q_\phi(z \mid x)$ are derived by the encoder.

Given an inference model $q_\phi(z \mid x)$, the evidence lower bound (ELBO) can be derived as follows:

$$\log p_\theta(x) = E_{q_\phi(z|x)}[\log p_\theta(x)] \tag{5}$$

$$= E_{q_\phi(z|x)}[\log \frac{p_\theta(x \mid z)\, p_\theta(z)}{p_\theta(z \mid x)}] \tag{6}$$

$$= E_{q_\phi(z|x)}[\log \frac{p_\theta(x \mid z)\, p_\theta(z)}{p_\theta(z \mid x)} \frac{q_\phi(z \mid x)}{q_\phi(z \mid x)}] \tag{7}$$

$$= E_{q_\phi(z|x)}\left[\log p_\theta(x \mid z) + p_\theta(z) - \log q_\phi(z \mid x)\right] + D_{KL}(q_\phi(z \mid x) \,||\, p_\theta(z \mid x)) \tag{8}$$

In Equation (8), the first term is ELBO, and the second term is the Kullback-Leibler (KL) divergence of the approximate $q_\phi(z|x)$ from the true posterior $p_\theta(z|x)$. To ensure $q_\phi(z|x)$ gets closer to $p_\theta(z|x)$, the KL divergence term between them has to be minimized. According to the equation, minimizing KL divergence can be transformed into the task of maximizing ELBO. Therefore, the loss function of the VAE can be expressed as follows:

$$L_{VAE}(\theta, \phi, x) = -E_{q_\phi(z|x)}[\log p_\theta(x \mid z) + \log p_\theta(z) - \log q_\phi(z|x)] \tag{9}$$

The VAE has been successfully applied in different domains. With a sliding window, the VAE can be used for the clustering of wafer map patterns [49]. However, the standard VAE with CNN is not used to classify wafer map defect patterns. Hence, the standard VAE needs to be modified to identify wafer map defect patterns by addressing imbalanced data problems.

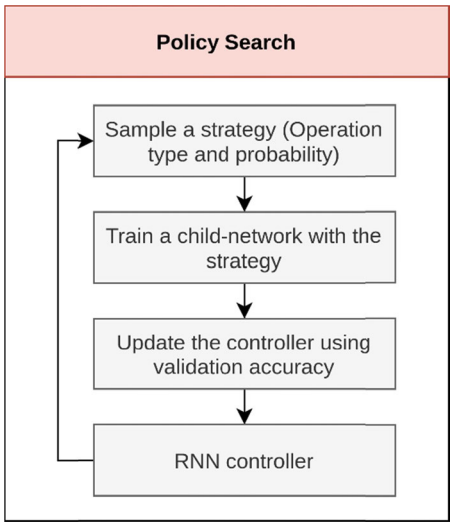

FIGURE 3. Overview of policy search.

3) POLICY SEARCH

We formulate the problem of finding the best augmentation policy as a discrete search problem. The operations we searched were rotation (5, 10, 15, 20, 25, 30, 35, 40, 45), flipping (horizontal and vertical), shifting (width and height), shearing range (horizontal and vertical), and zooming (1%-20%). In total, we have 46 operations in the search space.

The search algorithm used in our experiment uses Reinforcement Learning, inspired by [50]–[54]. The search algorithm has two components: a controller, which is a recurrent neural network, and a training algorithm, which is a proximal policy optimization algorithm [55]. At each step, the controller predicts a decision produced by a softmax, and the prediction is then fed into the next step as an embedment. In total, the controller has 46 softmax predictions to predict policies, each requiring an operation type and probability. The controller is trained with a reward signal, which is how good the policy is in improving the generalization of a ''child model'' (a neural network trained as part of the search process). In our experiments, we set aside a validation set to measure the generalization of the child model. A child model is trained using the augmented data generated by applying the policies on the training set. For each example in the mini-batch, one of the policies was chosen randomly to augment the image. The child model was used as a reward signal to train the recurrent network controller. As shown in Fig 3, the RNN controller predicts an augmentation policy from the search space. A child network with a fixed architecture was trained to attain convergence, achieving accuracy. The reward is used, with the policy gradient method, to update the controller so that it can generate better policies over time.

4) NEURAL NETWORK CLASSIFIER

To establish a predictive model, we employ a simple NN classifier followed by the downstream of the CVAE, which fine-tunes the CVAE encoder part ($f^{CVAE(encoder)}$) and feature extraction layers in an end-to-end manner for the

identification task of wafer map defect patterns. The predictor function (f^{NN}) can be summarized in Equation (10) as follows:

$$y' = f^{NN}(f^{CVAE(encoder)}(x)) \tag{10}$$

The objective function of the NN classifier is to predict the true class labels to minimize the cross-entropy loss between the approximate distribution and the ground truth distribution. The objective function of the predictor network (classification loss) is summarized as shown in Equation (11):

$$L_{NN}(x) = \sum y \log y' \tag{11}$$

where y is the ground truth value, and predicted y' is the predicted value.

The supervised NN classifier network provides predictions of wafer map defect patterns as any of the given defect patterns or non-defects.

B. TRAINING

To train a CNN model directly, we need large-scale image data such as the WM-811K dataset [56], which contains more than a hundred thousand images, but it is highly imbalanced. If large-scale training data are required, the applicable problems of a CNN are very limited. To avoid such situations and to make a CNN effective even for small-scale data, two important steps have been performed sequentially. The first step is to pre-train the generative models and replay the data samples for downstream tasks. The second step is to fine-tune the encoder of the pre-trained model, followed by a supervised classifier to perform the prediction.

1) GENERATIVE PRE-TRAINING

During training, the gradients of the loss function are required for the optimization of the ELBO. However, it is not easy to differentiate the loss with respect to the variational parameters ϕ because the gradients cannot be back propagated through the latent variable z. Hence, the re-parameterization trick, following the work in [57], is applied to overcome this problem.

The latent variable z is assumed to be a deterministic function of x and a random variable ε sampled from a fixed distribution, $N(0, 1)$. Hence, the non-differentiable random variable z is converted to a differentiable function of x and a random ε.

$$z = \mu_z + \sigma_z \odot \varepsilon, \varepsilon \sim N(0, 1) \tag{12}$$

where μ_z and σ_z are the variational parameters derived from the encoder. The sampling number L during the training was set to 1 because one sample was already sufficient. With model loss, the negative ELBO, we trained the model using the Adam optimizer [58] to update the weightings of the model.

2) FINE-TUNING FOR CLASSIFICATION

Fine-tuning involves tuning the parameters pre-trained with large-scale data using small-scale data. We fine-tuned the encoder of the pre-trained CVAE, pre-trained with an imbalanced large amount of data. We added a supervised NN classifier after the encoder of the CVAE, ignoring the decoder part. With model loss and cross-entropy, we also trained the model using the Adam optimizer [58] to update the weightings of the model.

TABLE 1. Model parameters.

Layer	Input Size	Input Channel	Filter size
Input	224x224	1	-
Conv2D	222x222	16	3x3
MaxPool	111x111	16	2x2
Conv2D	111x111	16	3x3
Conv2D	111x111	32	3x3
MaxPool	55x55	32	2x2
Conv2D	55x55	32	3x3
Conv2D	55x55	64	3x3
MaxPool	27x27	64	2x2
Conv2D	27x27	64	3x3
Conv2D	27x27	128	3x3
MaxPool	13x13	128	2x2
Conv2D	13x13	128	3x3
Mean	1x1	512	-
Std.Dev	1x1	512	-
Dense	1x1	512	-
Transformation	1x1	512	-
Dense	1x1	512	-
Dense	1x1	512	-
Dropout	1x1	512	-
Softmax	1x1	9	-

C. HYPERPARAMETERS

In this study, we constructed a CNN-based VAE model for WMDP, which has an encoder and decoder, each consisting of one input layer, eight convolution layers each with batch normalization, padding, and rectified linear unit (ReLU) activation, and five pooling layers (four stacking pairs of convolution-pooling-convolution). The supervised classification layer has one dropout layer, two fully connected layers, and one output layer. For a fair comparison, we used the same convolution-based neural network architecture for all the methods. In this model, each convolution and pooling layer consists of subsampling filters of size 3×3 and 2×2, respectively.

The first convolution layer extracts the features from the input training wafer images of size 224×224 pixels. Each convolution layer contained a set of learnable filters to extract unique feature maps. The number of filters increases with increasing depth of the convolution layer, and thus the number of feature maps also increases. However, feature maps become smaller and more complex due to the pooling layer in a deeper network. The proposed CNN-WDI model adopts 16, 32, 64, and 128 feature maps for the first, second, third, and fourth stacking pairs, respectively. The model parameters used in this study are listed in Table 1.

Zero padding was applied to all convolutional layers to ensure that the dimensions of the input and output feature

maps were the same. The Softmax activation function was applied to the output layer of the model. In addition, the Adam optimization method, which combines the concepts of Momentum optimization and root mean squared prop (RMSProp), was selected as the optimizer. This optimizer helps achieve a higher accuracy and improves the training process. In addition, after many attempts, other parameters such as batch size, learning rate, and number of pre-training and training epochs were assigned as 128, 0.001, 500, and 20, respectively. A smaller batch size improves the generalization ability by computing an approximation of the gradient value and then updating the other parameters.

IV. EXPERIMENTS

In this section, we first describe the experimental dataset used in this study. Then, we show the metrics used for evaluating all the methods. Finally, we provide the comprehensive experimental results.

A. DATASET

The WM-811K dataset is a semiconductor dataset consisting of 811,457 real wafer map images [56]. The wafer images were collected from 46,293 lots in a circuit probe test of the semiconductor fabrication process. A single lot contains 25 wafer maps, so there should be 1,157,325 wafer maps in total (i.e., 46,293 lots $\times$ 25 wafer/lot). Not all lots have exactly 25 WMs, due to sensor faults or other unknown reasons, and they were pruned from the dataset. The dataset also contains additional information about each wafer map, such as lot name, die size, wafer index number, failure type, and training and test labels. This is the largest publicly available wafer map dataset that can be accessed on the Multimedia Information Retrieval (MIR) laboratory website [59]. Different sizes of wafer images exist because of their two-dimensional nature and different pixel values along the length and width of the image. We found a total of 632 wafer images of various sizes ranging from 6×21 to 300×202.

Domain experts were responsible for defining nine different defect classes of wafer maps and assigning manual labels to 172,950 (21.3%) wafer maps in the entire dataset. Unfortunately, the labeled dataset is highly imbalanced, and only the no-defect class occupies 147,431 (85.2%) wafer maps of the labeled dataset. The other eight defect classes, that contain 25,519 (14.8%) wafer maps of the labeled dataset in total, are given as Edge-Ring: 9680 (5.6%), Edge-Local: 5189 (3.0%), Center: 4294 (2.5%), Local: 3593 (2.1%), Scratch: 1193 (0.7%), Random: 866 (0.5%), Donut: 555 (0.3%), and Near-full: 149 (0.1%). Fig 4 shows the randomly selected wafer defect images from each class.

We split the experimental dataset into training, validation, and testing sets, as shown in Table 2.

B. EVALUATION MEASURES

The measurements obtained from the confusion matrix were compared with the classification achievements, obtained from sentiment classification in similar studies, to demonstrate the accuracy of the method. Accuracy, precision, recall, and F1 measurement values were obtained from the confusion matrix.

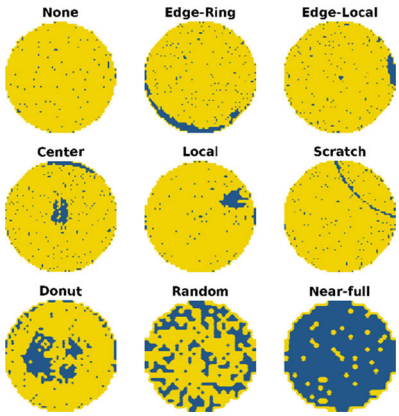

FIGURE 4. Typical examples of nine wafer defect classes.

TABLE 2. Experimental dataset.

	Train	Val	Test	Total
None	106,074	11,760	29,597	147,431
Edge-Ring	7,043	787	1,850	9,680
Edge-Local	3,796	414	979	5,189
Center	3,064	374	856	4,294
Local	2,557	274	762	3,593
Scratch	819	116	258	1,193
Donut	419	37	99	555
Random	647	64	155	866
Near-Full	105	10	34	149

The abbreviations TP (true positive), FP (false positive), FN (false negative), and TN (true negative) in the confusion matrix in Table 1 have the following meanings:

The accuracy, precision, recall, and F1 measurement were calculated according to the confusion matrix in Table 1. The accuracy was calculated according to Equation (13).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

Precision is the total estimate of class labels accurately predicted for each class. The precision was calculated using Equation (14).

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

The recall value is the weighted average of the correct labels that are correctly classified for each class. This value was calculated according to Equation (15):

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Other metrics, F1, were used to combine the precision and recall values in a single measurement. The value of this measurement is between 0 and 1, and if the classifier correctly classifies all samples, it takes the value of 1. The F1 measure

is given in Equation (16), and the F1 value is close to 1 for good classification success.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

All experiments were executed on an Intel Xeon E5-2698 v4 @ 2.20GHz, 256GB (CPU), NVIDIA Tesla V100 32GB (GPU), and Ubuntu 18.04 operation system. We also used the Scikit-Learn and Pytorch libraries with the Python programming language for all analyses.

C. RESULTS AND DISCUSSIONS

In this section, we present some experimental results, including a feature analysis that is selected by the CVAE. We then discuss a comparative analysis with other baseline methods and the efficiency of the proposed method.

1) GENERATION OF DEFECT PATTERNS

First, we pre-trained the unsupervised CVAE model on the entire training set and corroborated it using the validation set, as discussed previously. A CNN was used to extract visual features, and VAE was used to learn the distribution of each class label. We attempted to minimize the reconstruction loss (mean squared error) during training on the training set. The reconstruction error for 500 epochs in the training set is shown in Fig 5. It constantly decreases, and it shows the learning capability of our pre-trained model. The mean squared error was used as the reconstruction error in our experiment.

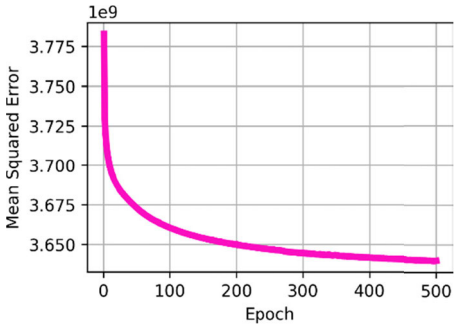

FIGURE 5. Reconstruction loss.

During training, we also tried to find the optimal augmentation policy, composed of several image processing operations such as rotation, flipping, shifting, shearing range, and zooming. As shown in Fig 6, we illustrated the examples of each operation applied to the generated samples.

As shown in the figure, the generated images were automatically transformed by image processing operations instead of using manual data augmentation. We used the rotation range from 5 to 45 degree and horizontal and vertical flipping. These transformations do not change the size of the generated images. In contrast, the other transformations such as shifting, shearing, and zooming change the size of generated images. For example, we used the zooming by between 1% and 20%. The hybrid method sequentially integrated image generation and various transformations can also address the data imbalance problem efficiently.

TABLE 3. Performance comparison.

	Precision	Recall	F1
With manual augmentation			
SVM [60]	87.5	91.0	88.0
ANN [61]	95.2	95.9	95.4
VGG-16 [62]	80.3	80.1	79.9
CNN-SD [18]	94.8	94.8	94.8
CNN-BN [18]	95.6	95.6	95.6
CNN-D [18]	95.2	95.2	95.2
CNN-WDI [18]	96.2	96.2	96.2
Without manual augmentation			
CNN-WDI [18]	90.3	86.4	87.7
CVAE (without image transformation)	91.7	94.4	93.3
CVAE (with image transformation)	93.6	96.9	95.1

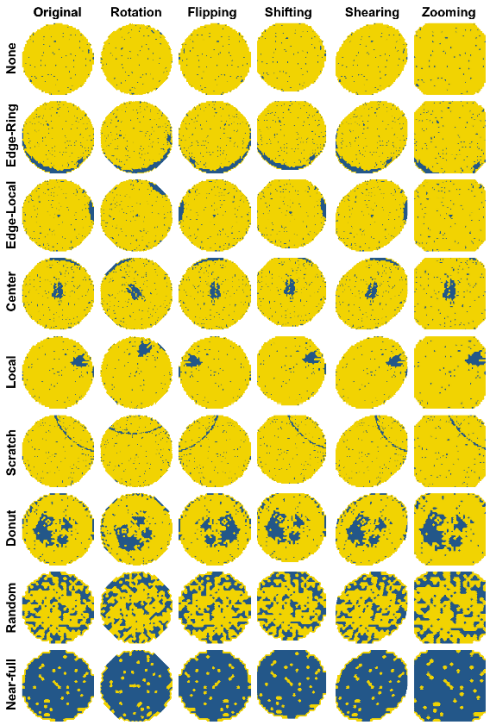

FIGURE 6. Generation of defect patterns.

2) PERFORMANCE EVALUATION

Secondly, we fine-tuned the only encoder part followed by a simple neural network classifier for the identification task of WMDP. We trained the supervised classifier on the training dataset and evaluated it on the validation set. We attempted to minimize cross-entropy loss during training. During training, the classification loss was constantly decreasing among all 20 epochs.

TABLE 4. Confusion matrix.

Labels	None	Edge-Ring	Edge-Local	Center	Local	Scratch	Donut	Random	Near-full
None	29,592 (99.98%)	0	0	2	0	0	3	0	0
Edge-Ring	3	1835 (99.19%)	0	0	1	4	4	0	3
Edge-Local	0	0	965 (98.57%)	3	4	0	4	3	0
Center	4	0	0	843 (98.48%)	1	0	3	4	1
Local	0	3	2	1	748 (98.16%)	0	3	3	2
Scratch	0	0	2	1	0	249 (96.51%)	3	1	2
Donut	1	3	1	1	2	0	89 (89.90%)	0	2
Random	0	2	1	3	0	1	0	146 (94.19%)	2
Near-full	1	0	0	0	0	0	0	0	33 (97.06%)

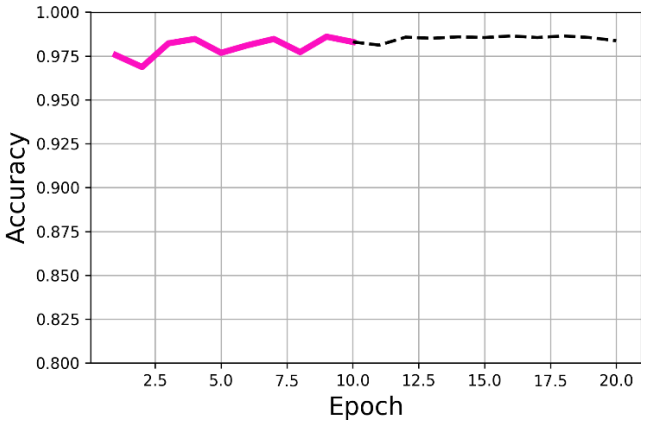

FIGURE 7. Accuracy of our proposed method on the validation set.

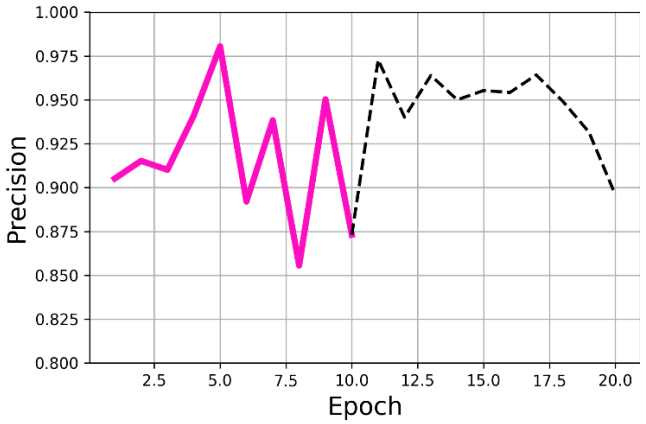

FIGURE 8. Precision of our proposed method on the validation set.

We evaluated the proposed method on the validation set standard measures such as accuracy, precision, recall, and F1-score. The classification performances on the validation set is shown in Fig 7-10, respectively. We achieved satisfying results in the first ten epochs. We highlighted the first ten and last ten epochs as solid pink and dashed black lines, respectively. We could not get clear information from the accuracy (Fig 7) for the imbalanced dataset. As you can see, we achieved the highest precision of 98.05% at the 5 th epoch (Fig 8) and the highest recall of 96.83% at the 8 th epoch (Fig 9). Our model has been satisfied at the 9th epoch by achieving the F1-score of 95.82% (Fig 10).

We compared the proposed methods to the other baseline methods such as SVM [60], ANN [61], VGG-16 [62], and CNN-WDI [18] algorithms. For fair comparison on the different split of the testing dataset. In the previous works, CNN-WDI [18] shows the highest performance results. We re-implemented the CNN-WDI method that achieved the comparative results as shown in Table 3. As shown in this table, the methods with manual data augmentation show high results. In this paper, we develop an automatic WMDP identification method without any manual augmentation. Because manual data augmentation is very time-consuming and non-memory efficient. Our hybrid method with the generative model and automatic image transformation operations can reduce the memory usages and much human efforts. We developed the CVAE method without any image transformation by only generating data samples. It improved the classification performance by 6%. Then we applied automatic image transformation with policy search strategy, to the CVAE method. It shows the highest classification performance without manual data augmentation and comparative results with manual data augmentation techniques. As conclude, the experimental results shown in Table 3 highlights the efficiency of our proposed method. As shown, Saqlain *et al.* [47] achieved the F1-score of 87.7% on the

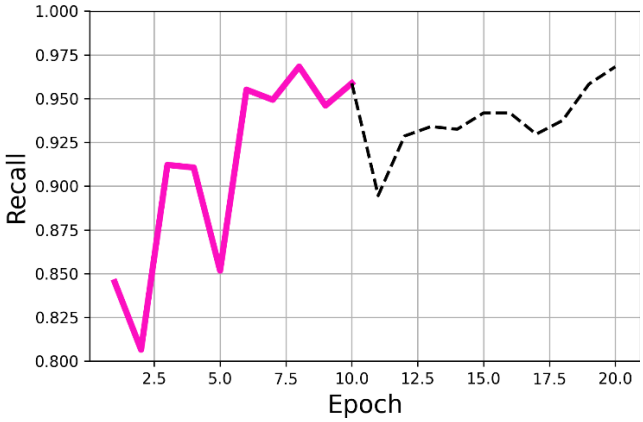

FIGURE 9. Recall of our proposed method on the validation set.

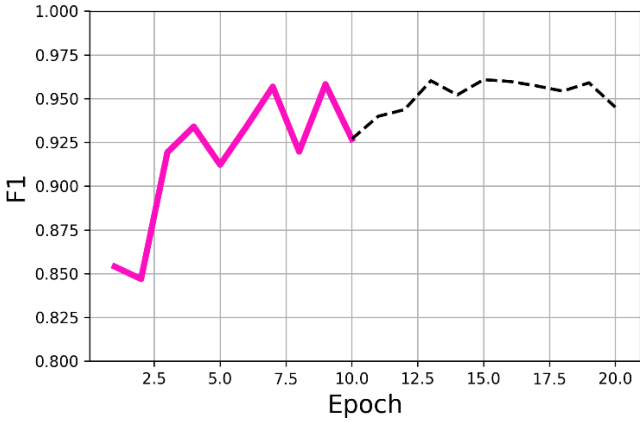

FIGURE 10. F1-score of our proposed method on the validation set.

original imbalanced data and achieved the F1-score of 96.2% on the manually balanced data. Our proposed method, CVAE with automatic image transformation with policy search strategy, achieved the F1-score of 95.1% without any human efforts. Surprisingly, the proposed CVAE method achieves the highest recall of 96.9%. It is very comparative to the manual augmentation methods in terms of predictive performance and can reduce much human effort.

As shown in Table 4, the confusion matrix performed by our proposed method CVAE with image transformation is provided. As you can see, we achieved high accuracy results higher than 90% except for Donut defect pattern.

In this paper, we addressed the issue of manual data augmentation; it requires much human effort. Instead of manually transforming training data, we automatically generated fake data similar to original images and added an image transformation function with a policy search strategy. For a fair comparison, we selected the same image transformation techniques used in the previous works. It reduces many preprocessing steps and immensely scalable to add more image transformation techniques. As shown in Table 3, the proposed method CVAE is lower than the performance of the highest manually augmented method. However, we can quickly improve it by adding other image transformation techniques. The policy search algorithm is very efficient in finding the best augmentation policy from many possible states even there are many transformation techniques. May it increases the computation time and memory usage. But it is not critical in this research, and we can reduce it at the application level for real-world scenarios.

V. CONCLUSION

In this study, we developed a DL-based method, that is, CVAE for WMDP, which employs CNN as a feature extractor, and CVAE exploits the full connection between the features and the subsequent convolved images in an unsupervised manner. A simple NN classifier was used to identify the defect patterns from input images in a supervised manner. The robust and discriminative features from the wafer map through this network can be extracted to identify the WMDP improvement. Additionally, an automatic policy search procedure was defined for improved data augmentation, instead of using manual functions. CVAE achieves better recognition results on real-world wafer map datasets than traditional WMDP methods and other DL models. The comprehensive experimental results verify that the CVAE is capable of learning effective features from wafer maps. This study provides a new method for the identification of WMDP using generative DL models, with an automatic data augmentation procedure, control. It addresses the problem of data imbalance and limited training data, which leads to overfitting of DL-based methods.

The limitations of the proposed method are described as follows. In the general research of wafer map defect pattern, most methods utilized the limited dataset publicly available. More challenging data is necessary to this semiconductor manufacturing research field. We proposed automatic techniques such as generative model and image transformation with the policy search strategy to reduce human efforts. However, it improves the computational cost, but it can be reduced. We only considered the five transformations in the image transformation phase, such as rotation, flipping, shifting, shearing range, and zooming. There is also not exact value of augmented data size for training.

In the future, we will discover more data that covers more challenging issues in this research field. Also, we will carry out further research on other generative models, that is, generative adversarial networks and improved deep network architecture to disclose the properties of CVAE. Additionally, fast and adaptive algorithms for searching data augmentation policies will be considered. We will improve the proposed method in terms of both computational cost and predictive performance for developing real-world applications. To increase the capability, we will employ more image transformation techniques and discover augmented data characteristics.

ACKNOWLEDGMENT

(Ho Sun Shon and Erdenebileg Batbaatar contributed equally to this work.)

REFERENCES

[1] L. Jelinek, "Global semiconductor market trends," IHS Markit, London, U.K., Tech. Rep. CMP7'18-1 IHS, May 2018. [Online]. Available: https://nccavs-usergroups.avs.org/wp-content/uploads/CMPUG2018/CMP718-1-IHS.pdf

[2] G. Batra, *Artificial-Intelligence Hardware: New Opportunities for Semiconductor Companies*. New York, NY, USA: McKinsey & Company, Jan. 2018.

[3] N. Shin, K. L. Kraemer, and J. Dedrick, "R&D and firm performance in the semiconductor industry," *Ind. Innov.*, vol. 24, no. 3, pp. 280–297, Apr. 2017, doi: 10.1080/13662716.2016.1224708.

[4] L. Mönch, 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, Jul. 2018, doi: 10.1080/00207543.2017.1401233.

[5] R. Uzsoy, J. W. Fowler, and L. Mönch, "A survey of semiconductor supply chain models part II: Demand planning, inventory management, and capacity planning," *Int. J. Prod. Res.*, vol. 56, no. 13, pp. 4546–4564, Jul. 2018, doi: 10.1080/00207543.2018.1424363.

[6] L. Mönch, R. Uzsoy, and J. W. Fowler, "A survey of semiconductor supply chain models part III: Master planning, production planning, and demand fulfilment," *Int. J. Prod. Res.*, vol. 56, no. 13, pp. 4565–4584, Jul. 2018, doi: 10.1080/00207543.2017.1401234.

[7] C.-C. Hsieh, C.-Y. Liu, P.-Y. Wu, A.-P. Jeng, R.-G. Wang, and C.-C. Chou, "Building information modeling services reuse for facility management for semiconductor fabrication plants," *Autom. Construct.*, vol. 102, pp. 270–287, Jun. 2019, doi: 10.1016/j.autcon.2018.12.023.

[8] Y.-T. Kao, S. Dauzère-Pérès, J. Blue, and S.-C. Chang, "Impact of integrating equipment health in production scheduling for semiconductor fabrication," *Comput. Ind. Eng.*, vol. 120, pp. 450–459, Jun. 2018, doi: 10.1016/j.cie.2018.04.053.

[9] J. D. Mohn, "System and apparatus for flowable deposition in semiconductor fabrication," U.S. Patent 9 719 169, Aug. 1, 2017.

[10] N. Dimitriou, L. Leontaris, T. Vafeiadis, D. Ioannidis, T. Wotherspoon, G. Tinker, and D. Tzovaras, "A deep learning framework for simulation and defect prediction applied in microelectronics," *Simul. Modell. Pract. Theor.*, vol. 100, Apr. 2020, Art. no. 102063, doi: 10.1016/j.simpat.2019.102063.

[11] J. Wang, P. Fu, and R. X. Gao, "Machine vision intelligence for product defect inspection based on deep learning and Hough transform," *J. Manuf. Syst.*, vol. 51, pp. 52–60, Apr. 2019, doi: 10.1016/j.jmsy.2019.03.002.

[12] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," *IEEE Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 315–322, May 2018.

[13] L.-C. Wang, A. Wang, and C.-Y. Chueh, "Development of a capacity analysis and planning simulation model for semiconductor fabrication," *Int. J. Adv. Manuf. Technol.*, vol. 99, nos. 1–4, pp. 37–52, Oct. 2018, doi: 10.1007/s00170-016-9089-z.

[14] W. M. Zhong, "Die sorting apparatus and method," *ASM Assembly Automat.*, vol. 7, no. 345, p. 254, Mar. 2008.

[15] M. Breton, R. Chao, G. R. Muthinti, A. A. de la Peña, J. Simon, A. J. Cepler, M. Sendelbach, J. Gaudiello, S. Emans, M. Shifrin, Y. Etzioni, R. Urenski, and W. T. Lee, "Electrical test prediction using hybrid metrology and machine learning," *Proc. SPIE*, vol. 10145, Apr. 2017, Art. no. 1014504, doi: 10.1117/12.2261091.

[16] N. Kwon, H. Kang, and Y. Kim, "Semiconductor defect classification device, method for classifying defect of semiconductor, and semiconductor defect classification system," U.S. Patent 10 713 778, Jul. 14, 2020.

[17] M. Salem, S. Taheri, and J.-S. Yuan, "An experimental evaluation of fault diagnosis from imbalanced and incomplete data for smart semiconductor manufacturing," *Big Data Cognit. Comput.*, vol. 2, no. 4, p. 30, Sep. 2018, doi: 10.3390/bdcc2040030.

[18] M. Saqlain, Q. Abbas, and J. Y. Lee, "A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 436–444, Aug. 2020, doi: 10.1109/TSM.2020.2994357.

[19] 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, May 2013, doi: 10.1016/j.engappai.2012.11.009.

[20] Y. Liu and S. Zhou, "Detecting point pattern of multiple line segments using Hough transformation," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 13–24, Feb. 2015, doi: 10.1109/TSM.2014.2385600.

[21] Q. Zhou, L. Zeng, and S. Zhou, "Statistical detection of defect patterns using Hough transform," *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 3, pp. 370–380, Aug. 2010, doi: 10.1109/TSM.2010.2048959.

[22] Y.-S. Jeong, "Semiconductor wafer defect classification using support vector machine with weighted dynamic time warping kernel function," *Ind. Eng. Manage. Syst.*, vol. 16, no. 3, pp. 420–426, Sep. 2017, doi: 10.7232/iems.2017.16.3.420.

[23] Q. P. He and J. Wang, "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes," *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 345–354, Nov. 2007, doi: 10.1109/TSM.2007.907607.

[24] Q. P. He and J. Wang, "Principal component based k-nearest-neighbor rule for semiconductor process fault detection," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 1606–1611.

[25] B. Kim, Y.-S. Jeong, S. H. Tong, I.-K. Chang, and M.-K. Jeongyoung, "A regularized singular value decomposition-based approach for failure pattern classification on fail bit map in a DRAM wafer," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 41–49, Feb. 2015, doi: 10.1109/TSM.2014.2388192.

[26] 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, doi: 10.1109/TSM.2018.2806931.

[27] M. Saqlain, B. Jargalsaikhan, and J. Y. Lee, "A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 171–182, May 2019, doi: 10.1109/TSM.2019.2904306.

[28] 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, doi: 10.1109/TSM.2018.2795466.

[29] 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, doi: 10.1109/TSM.2018.2841416.

[30] S. Cheon, H. Lee, C. O. Kim, and S. H. Lee, "Convolutional neural network for wafer surface defect classification and the detection of unknown defect class," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 163–170, May 2019, doi: 10.1109/TSM.2019.2902657.

[31] C. H. Jin, H. J. Na, M. Piao, G. Pok, and K. H. Ryu, "A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 286–292, Aug. 2019, doi: 10.1109/TSM.2019.2916835.

[32] T. Ishida, I. Nitta, D. Fukuda, and Y. Kanazawa, "Deep learning-based wafer-map failure pattern recognition framework," in *Proc. 20th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2019, pp. 291–297.

[33] Z. Shen and J. Yu, "Wafer map defect recognition based on deep transfer learning," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2019, pp. 1568–1572.

[34] R. Wang and N. Chen, "Wafer map defect pattern recognition using rotation-invariant features," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 596–604, Nov. 2019.

[35] J. Yu, "Enhanced stacked denoising autoencoder-based feature learning for recognition of wafer map defects," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 613–624, Nov. 2019, doi: 10.1109/TSM.2019.2940334.

[36] Y. Yuan-Fu, "A deep learning model for identification of defect patterns in semiconductor wafer map," in *Proc. 30th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, May 2019, pp. 1–6.

[37] A. Shawon, M. O. Faruk, M. B. Habib, and A. M. Khan, "Silicon wafer map defect classification using deep convolutional neural network with data augmentation," in *Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2019, pp. 1995–1999.

[38] T. Nakazawa and D. V. Kulkarni, "Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder–decoder neural network architectures in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 2, pp. 250–256, May 2019, doi: 10.1109/TSM.2019.2897690.

[39] J. Yu, X. Zheng, and J. Liu, "Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map," *Comput. Ind.*, vol. 109, pp. 121–133, Aug. 2019, doi: 10.1016/j.compind.2019.04.015.

[40] J. Yu and J. Liu, "Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection," *IEEE Trans. Ind. Electron.*, early access, Aug. 6, 2021, doi: 10.1109/TIE.2020.3013492.

[41] M. B. Alawieh, D. Boning, and D. Z. Pan, "Wafer map defect patterns classification using deep selective learning," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.

[42] J. Jang, M. Seo, and C. O. Kim, "Support weighted ensemble model for open set recognition of wafer map defects," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 4, pp. 635–643, Nov. 2020, doi: 10.1109/TSM.2020.3012183.
[43] T.-H. Tsai and Y.-C. Lee, "A light-weight neural network for wafer map classification based on data augmentation," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 4, pp. 663–672, Nov. 2020, doi: 10.1109/TSM.2020.3013004.
[44] T.-H. Tsai and Y.-C. Lee, "Wafer map defect classification with depth-wise separable convolutions," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2020, pp. 1–3.
[45] J. Yu, Z. Shen, and X. Zheng, "Joint feature and label adversarial network for wafer map defect recognition," *IEEE Trans. Autom. Sci. Eng.*, early access, Jun. 30, 2020, doi: 10.1109/TASE.2020.3003124.
[46] C. H. Jin, H.-J. Kim, Y. Piao, M. Li, and M. Piao, "Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes," *J. Intell. Manuf.*, vol. 31, no. 8, pp. 1861–1875, Dec. 2020, doi: 10.1007/s10845-020-01540-x.
[47] R. Wang and N. Chen, "Defect pattern recognition on wafers using convolutional neural networks," *Qual. Rel. Eng. Int.*, vol. 36, no. 4, pp. 1245–1257, Jun. 2020, doi: 10.1002/qre.2627.
[48] 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, May 2017, doi: 10.1109/TSM.2017.2676245.
[49] J. Hwang and H. Kim, "Variational deep clustering of wafer map patterns," *IEEE Trans. Semicond. Manuf.*, vol. 33, no. 3, pp. 466–475, Aug. 2020, doi: 10.1109/TSM.2020.3004483.
[50] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," 2016, *arXiv:1611.02167*. [Online]. Available: http://arxiv.org/abs/1611.02167
[51] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 459–468.
[52] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.
[53] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation policies from data," 2018, *arXiv:1805.09501*. [Online]. Available: http://arxiv.org/abs/1805.09501
[54] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.
[55] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347
[56] 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.
[57] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[59] mirlab.org. (2018). *MIR Corpora*. [Online]. Available: http://mirlab.org/dataSet/public/
[60] W. S. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006, doi: 10.1038/nbt1206-1565.
[61] B. Yegnanarayana, *Artificial Neural Networks*. PHI Learning Pvt. Ltd., Jan. 2009.
[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

HO SUN SHON received the B.S. and M.S. degrees in statistics from Sungshin Women University, Seoul, South Korea, in 1986 and 1992, respectively, and the Ph.D. degree in computer science from Chungbuk National University, Cheongju, South Korea, in 2010. She is currently a Researcher with the Research Institute for Computer and Information Communication, Chungbuk National University. Her research interests include machine learning, data mining, pattern recognition, and bioinformatics.

ERDENEBILEG BATBAATAR received the M.S. and Ph.D. degrees in data mining, medical informatics, and computer science from the Database and Bioinformatics Laboratory, Chungbuk National University, South Korea. He is currently a Postdoctoral Researcher of bioinformatics and computer science with Chungbuk National University. His research interests include software engineering, data mining, big data analysis, bioinformatics, machine learning, deep learning, and their applications.

WAN-SUP CHO received the B.S. degree from Kyeongbuk National University, in 1985, and the M.S. and Ph.D. degrees from KAIST, South Korea, in 1987 and 1996, respectively. He is currently a Professor with the Department of Management Information Systems, Chungbuk National University. His research interests include big data platform and data governance with AI and the IoT for smart factories and smart healthcare.

SEONG GON CHOI received the B.S. degree in electronics engineering from Kyeongbuk National University, in 1990, and the M.S. and Ph.D. degrees from Information Communications University, South Korea, in 1999 and 2004, respectively. He is currently a Professor with the College of Electrical and Computer Engineering, Chungbuk National University. His research interests include smart grid, the IoT, mobile communication, high-speed network architecture and protocol.