A Variational Autoencoder Enhanced Deep Learning Model for Wafer Defect Imbalanced Classification

Shuyu Wang, Zhitao Zhong, *Student Member, IEEE*, Yuliang Zhao, and Lei Zuo

***Abstract*— In semiconductor foundries, wafer map defect analysis is crucial to prevent yield excursion. However, traditional manual inspection can hardly meet the high-throughput demand. Deep learning-based automatic defect detection shows promising efforts to achieve high accuracy and efficiency, yet the current approaches' performance is limited by the imbalanced dataset and lack of interpretability. In this article, we propose a variational autoencoder-enhanced deep learning model (VAEDLM) for wafer defect imbalanced classification. It is light-weighted and effective in wafer defect pattern recognition on imbalanced dataset. It used variational autoencoders (VAEs) and decoders to generate similar wafer defect maps and a refined deep convolutional neural network (CNN) for feature learning. We demonstrate the method using an authentic wafer map dataset, WM-811K. The performance is not only significantly improved after data augmentation, but it also beats the state-of-the art methods, reaching 99.19% accuracy, 99.10% recall, 99.23% precision, 99.96% AUC, and 99.16% for F1-score. It clearly demonstrates the method's efficacy to deal with the imbalanced defect pattern. Our study using saliency map and t-distributed stochastic neighbor embedding (t-SNE) further leads to enhanced interpretability.**

***Index Terms*— Convolutional neural network (CNN), data augmentation, encoder–decoder, pattern recognition, wafer defect.**

I. Introduction

INTEGRATED circuits (ICs) manufacturing is a highly complex process that involves hundreds of intricate steps. Therefore, the defect can be easily introduced into the wafer manufacturing process. These defects not only might result in reliability issues, but it also can be the main cause of low yield [1], [2]. Defects generated during fabrication can be deemed as the superposition of global defects (particle related) and local defects (process related). Global defects arise from random causes, such as scattered particles on the wafer. Such defects might relate to the clean room and it is expensive to amend. Local defects are normally associated with human mistakes or chemical stains. Typical examples are line scratch, edge, ring, blob, and bull's-eye types of patterns [3].

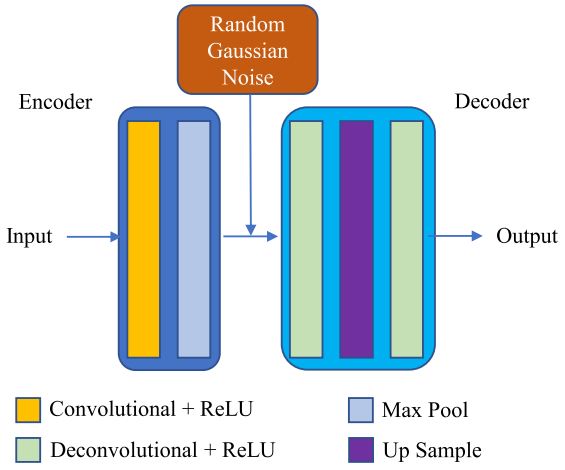

Fig. 1. Structure of the autoencoder–decoder model.

An experienced engineer can identify a potential root cause based on the defect pattern analysis to prevent a yield excursion. For example, a cluster with a curvilinear shape is probably caused by a material handling scratch. The scratch type may be related to poor wafer handling. Edge and ring failures may be caused by etch rate homogeneity issues. Center type may be caused by a nonuniform temperature distribution across the wafer during the rapid thermal annealing process. These patterns on wafer maps provide important clues on which step of the manufacturing process is responsible for the failures [4], [5]. As a result, the defect analysis and immediate process adjustment are crucial for yield improvement. Meanwhile, the wafer production amount in foundries is on the scale of ten million, which demands high-throughput wafer defect analysis methods. Traditional manual review by engineers is low productive despite high accuracy. Automated defect scanning, on the other hand, can promptly scan wafer surfaces to identify the locations and sizes of the defects.

Many studies on IC manufacturing defect automatic detection based on machine learning emerged in recent years. They could be divided as unsupervised learning and supervised learning [6]. The branch of the supervised learning for defect classification used feature extraction-based approach for pattern recognition in the early days, where defects were first preprocessed with data transformation to measure several features of wafer surface images, such as the size, shape, location, and color of possible defects [5], [7] and then classified by machine learning techniques [8], [9]. Wu *et al.* [5] proposed a model, wafer map failure pattern recognition (WMFPR), to extract

Manuscript received June 22, 2021; accepted November 2, 2021. Date of publication November 8, 2021; date of current version December 16, 2021. The work of Shuyu Wang was supported in part by the Natural Science Foundation of China under Grant 62104034, in part by the Fundamental Research Fund from Central University under Grant 2023012, and in part by the Natural Science Foundation of Hebei Province under Grant F2020501033. Recommended for publication by Associate Editor Z. Zhou upon evaluation of reviewers' comments. *(Shuyu Wang and Zhitao Zhong contributed equally to this work.) (Corresponding author: Shuyu Wang.)*

Shuyu Wang, Zhitao Zhong, and Yuliang Zhao are with the School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China (e-mail: wangshuyu@neuq.edu.cn).

Lei Zuo is with the Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCPMT.2021.3126083.

Digital Object Identifier 10.1109/TCPMT.2021.3126083

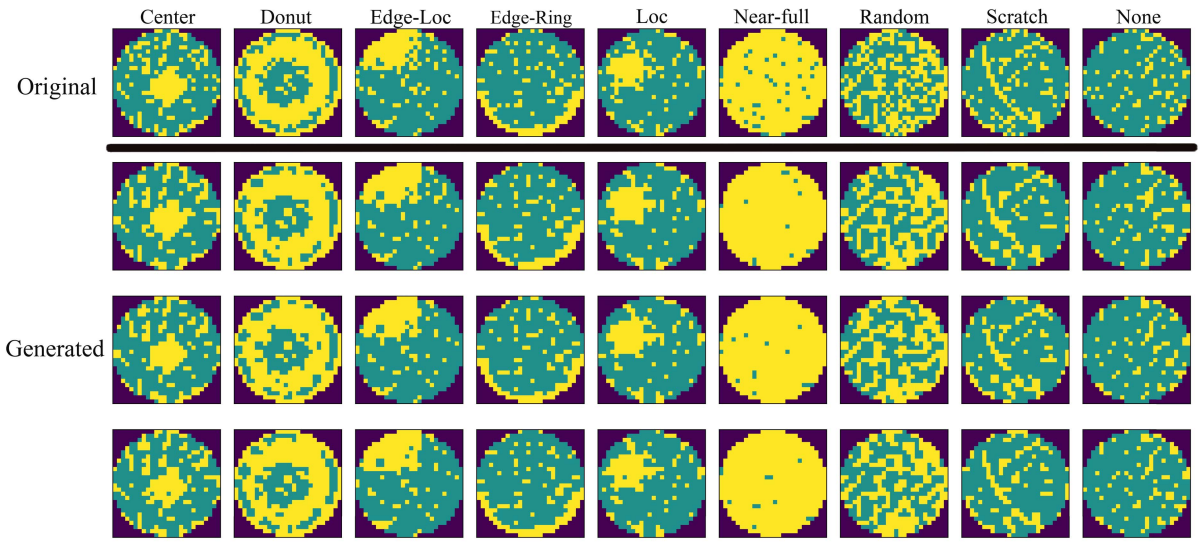

Fig. 2. Comparison of the original and generated wafer map.

features based on Radon transform and used a support vector machine (SVM) classifier to classify defects. Similarly, Piao *et al.* [10] applied the ensemble-based decision tree model to classify wafer defects after feature extraction. Saqlain *et al.* [11] used an ensemble-based classification mode, wafer map defect pattern identification (WMDPI). It combines random forest (RF), logistic regression (LR), SVM, and artificial neural network (ANN). These methods' performance relied on the feature engineering skills, which required complex domain-related expertise.

More recent researches used deep learning method to classify the wafer defect. In the past few years, deep learning gained tremendous success in image recognition field. It is an end-to-end method that can process the high-dimensional raw data without feature selection process. Nakazawa and Kulkarni [12] applied convolutional neural network (CNN) for wafer defect classification and showed the method to be highly effective. Lee *et al.* [13] used CNN to find multivariate process faults and diagnose the source of the faults. Kyeong and Kim [14] applied the CNN model to classify mixed-typed defect patterns of wafer bin map. Yu *et al.* [4] proposed a stacked convolutional sparse denoising auto-encoder (SCSDAE) for wafer map pattern recognition. It integrated CNN and sparse denoising autoencoder (SDAE) to learn effective features.

Despite the advancement of the automatic wafer defect detection, one major issue, the imbalanced wafer defect types, is still seriously deteriorating the classification performance. In a real wafer dataset, WM-811K, there are nearly 38 % wafer maps belong to the type "Edge-Ring," whereas less than 1 % wafer maps are "Near-Full." And this class imbalance issue is one of the main reasons for the low performance in the defect detection. It is easy to be interpreted as the model can learn the majority class more easily, yet harder to master the minority cases. Meanwhile, the learning ability of the classifiers to detect different pattern is not the same, and it is worthy of more study to have a deeper understanding into this problem. These reasons motivate researchers to solve dataset inequality issue with data augmentation. Saqlain *et al.* [15] augmented the dataset by random rotation, flipping, and shifting the pattern. Wang *et al.* [16] applied adaptive balancing Generative Adversarial Network to generate simulated wafer maps in high fidelity. However, it has not been adequately addressed and need further studies.

Another existing issue remains in the current deep learning approach is the lack of sufficient interpretability. The CNN method inherently has the black-box nature. How the data has been processed by the neural network for improved performance and how it captures the feature is not well understood. The interpretability study can guide the researchers to optimize the model's architecture and better evaluate the failed cases, yet it is still not fully examined in wafer defect detection.

In this study, we solve the wafer defect imbalanced classification problem by data augmentation with variational autoencoder (VAE). In this way, the amounts of various wafer defect are more balanced for proper training. We then use a deep CNN-based method to detect the defect types using the WM-811K dataset. The model evaluation results indicate that the method is highly effective to boost the performance and it outperforms the previous studies. Last, we used t-distributed stochastic neighbor embedding (t-SNE) and saliency map to interpret the feature learning process and visualize extracted features, so that it can guide us on the misclassification analysis and understand the detection process.

II. Method

A. Dataset

We used WM-811K dataset, which includes 8 11 457 wafer maps, provided by a real semiconductor foundry. The wafer maps are from 46 293 different lots, and each lot has 25 images. Human experts have annotated 1 72 950 of the total 8 11 457 data.

Wafer maps are divided into nine classes, including "Center," "Donut," "Edge-Loc," "Edge-Ring," "Loc," "Near-full," "Random," "Scratch," and "None." The "None" type is not annotated as a specific failure type. 25 519 of the annotated wafer maps have specific failure patterns, which is 14.76% of annotated data. In these specific failure pattern data, the

distribution of different types of failure pattern is extremely imbalanced. Center type has 4294 images. Donut has 555. Edge-Loc has 5189. Edge-Ring has 9680. Loc has 3593. Random has 866. Scratch has 1193, and Near-full has 149.

Each wafer map image is consisted of a 2-D array pixels, and every pixel have three possible values. "0" means the region outside the boundary of the wafer, "1" means normal region within the wafer boundary and "2" means this region is defect. We used one-hot encoding to transform the size of each wafer map from (width, height) to (width, height, 3), [1, 0, 0] corresponding to 0, [0, 1, 0] corresponding to 1, and [0, 0, 1] corresponding to 2. This process is as follows:

$$\begin{cases} N(w,h,i)=0, & F(w,h)\neq i \\ N(w,h,i)=1, & F(w,h)=i \end{cases}$$

where $F(w,h)$ refers to the value of wafer map before one-hot encoding, and $N(w,h,i)$ refers to the value of wafer map after one-hot encoding.

B. Data Augmentation

The distribution of these failure patterns is extremely imbalanced, which can deteriorate the performance, since some scarce failure patterns' feature may not be well trained. Data augmentation technology has been used in many classification tasks to solve the class imbalance problem. We generate more data using the autoencoder and decoder to make the defect types more balanced. The generated patterns are similar to the original images. In Fig. 1, we illustrate the schematic view of our autoencoder–decoder's structure. A convolution layer with 64 3 × 3 kernels and a max pooling layer forms the encoder. Two deconvolution layers with upsampling form the decoder. The convolution kernel uses stride 1 with the same padding. The max pool layer's pool size is 2 × 2. The first and second deconvolution layers use a 3 × 3 kernel with the same padding, and the numbers of kernels are 64 and 3, respectively. Both convolution layers and first deconvolution layers use the ReLU function as the activation function. The second deconvolution layer uses the sigmoid function as activation function. Random Gaussian noise is added between the encoder and the decoder. In this way, more images with similar patterns are generated and they are not the same as the original ones. Comparison of the original wafer maps and the generated wafer maps is shown in Fig. 2.

C. Proposed Deep Learning Model

Fig. 3 shows the schematic view of the deep learning model structure. The VAE-enhanced deep learning model (VAEDLM) is consisted of multiple basic blocks and pooling layers, and each block is composed of a convolution layer, using 64 3 × 3 convolution kernels and a stride of 2. The batch normalization [17] layer uses momentum = 0.99 and a leaky rectified linear unit [18] with a leaking rate of 0.01. The first block extracts the features of a one-hot encoded wafer map with a size of (26, 26, 3), then add a 2 × 2 max pooling layer.

In Fig. 3, the ⊕ sign refers to add operation, and the network uses a feedforward connection. It divides the input data into two branches. The main branch has two stacked blocks, and each block has two convolution layers with 64 3 × 3 kernels. The other branch used 64 1 × 1 kernels for convolution with a stride of 2. Next, the input data goes through another similar feedforward connection, which adds to the output of the main branch. By comparison, the convolution layer on the main branch use stride 1. A global average pooling layer is added before the last output layer, and it replaces the fully connect layer. Last, a softmax activation function will classify the data into nine types. In total, there are 1 55 529 parameters in the model. All convolution layers in this model use 0.0001 as the weight for L2 regularization.

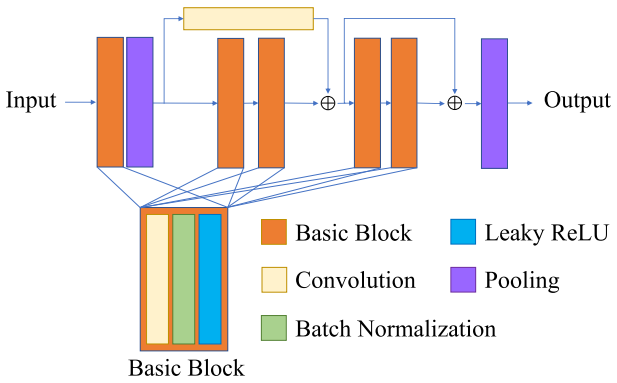

Fig. 3. Schematic view of the deep learning model's structure.

TABLE I
WAFER MAP AMOUNT CHANGES BEFORE AND AFTER DATA AUGMENTATION

Failure Pattern	Before Augmentation		After Augmentation	
	Amount	Percentage	Amount	Percentage
Center	4294	2.48%	2160	6.96%
Donut	555	0.32%	2101	6.77%
Edge-Loc	5189	3.00%	2368	7.63%
Edge-Ring	9680	5.60%	2108	6.80%
Loc	3593	2.08%	2376	7.66%
Near-full	866	0.50%	2112	6.81%
Random	1193	0.69%	2146	6.92%
Scratch	149	0.09%	2160	6.96%
None	147431	85.24%	13489	43.48%

D. Evaluation Metrics

Our method is evaluated by accuracy, recall, precision, area under curve (AUC), and F1-score. The accuracy is the percentage portion of the true-positive results over all predicted positive of truly predicted data to all predicted data. Recall indicates the cases and precision indicates percentage of true-positive over all actual positive samples. AUC means area under the receiver operating characteristic (ROC) curve. F1-score combines the recall and precision.

E. Model Training

We use categorical cross-entropy as the loss function to measure the predicted probability distribution. Batch normalization is added for normalization. The adaptive gradient algorithm [19] is applied as the optimizer to minimize the loss function.

TABLE II
PERFORMANCE EVALUATION OF THE MODEL TRAINED ON AUGMENTED DATA AND ORIGINAL DATA

	Model Trained on Augmented Data					Model Trained on Original Data				
Pattern	Accuracy	Recall	Precision	AUC	F1-Score	Accuracy	Recall	Precision	AUC	F1-Score
Center	1.0000	1.0000	1.0000	1.0000	1.0000	0.8713	0.8653	0.8789	0.9709	0.8721
Donut	1.0000	1.0000	1.0000	1.0000	1.0000	0.7899	0.7899	0.8034	0.9548	0.7966
Edge-Loc	0.9819	0.9797	0.9819	0.9983	0.9808	0.7033	0.6984	0.7096	0.9213	0.7040
Edge-Ring	0.9975	0.9975	0.9975	1.0000	0.9975	0.9518	0.9493	0.9542	0.9942	0.9517
Loc	0.9888	0.9888	0.9911	0.9999	0.9899	0.5691	0.5610	0.5726	0.8586	0.5667
Near-full	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Random	1.0000	1.0000	1.0000	1.0000	1.0000	0.8497	0.8497	0.8596	0.9624	0.8547
Scratch	1.0000	1.0000	1.0000	1.0000	1.0000	0.1048	0.1004	0.1027	0.6115	0.1015
None	0.9871	0.9853	0.9874	0.9993	0.9863	0.9919	0.9917	0.9923	0.9993	0.9920
All	**0.9919**	**0.9910**	**0.9923**	**0.9996**	**0.9916**	0.9620	0.9611	0.9634	0.9920	0.9622

The model was developed using Keras 2.3.1 and TensorFlow 2.2.0. A NVIDIA Tesla P100 GPU with 16-GB memory was engaged to accelerate the calculation. Since the early stopping strategy was adopted, the training time is 327 s for 790 epochs with 80% of the data as the training set. The stop condition is when the valid loss is less than 0.002 in 30 epochs.

III. RESULTS

This part first presents the data augmentation result and then demonstrate the model's performance using various metrics. We also compare the result with previous researches.

A. Data Augmentation

Before training, all the input images are preprocessed by one-hot encoding and then we use the autoencoder and decoder to generate images and balance the amount of the defect types. Table I shows how the data amount changes before and after augmentation. After data augmentation, the amount of the wafer maps is between 2101 and 2376, except the "None" type has 13 489 images. The "None" type images are in various forms and hard to train, so we use larger quantity.

B. Model Performance Evaluation

Table II shows the model training result before and after data augmentation. The performance after augmentation is significantly enhanced compared with the original data. The model gets a satisfactory performance of all patterns in every metric. Especially, the testing results of "Donut," "Edge-Loc," "Loc," and "Scratch" are greatly boosted from the original ones. Most of the failure patterns can reach over 99% of accuracy, and "Edge-Loc" has an accuracy of 98.19%. "Loc" has an accuracy of 98.88%, and the accuracy of 'None' is 98.71%, which is a high-performance result for wafer defect pattern recognition. The confusion matrix is shown as Fig. 4. Most predicted results match the target well. Besides, it also shows some mismatched cases. For example, 0.9% of "Edge-Loc" data is taken as "Edge-Ring" by mistake, and 0.7% of "None" data is mismatched as the "Loc" type.

A comparison of our model's performance with the previous researches is shown in Table III. These researches all used the same dataset, WM-811K. SCSDAE [4] is a deep learning-based model with stacked convolutional autoencoder. WMFPR [5] is an SVM-based model. DTE-FPR [10] is a decision tree-based model. SVE [11] combines various classifier, including LR, RFs, gradient boosting machine (GBM), and ANNs. Then, it integrates the results as the output by soft voting, and all the above three are based on the feature extraction technology.

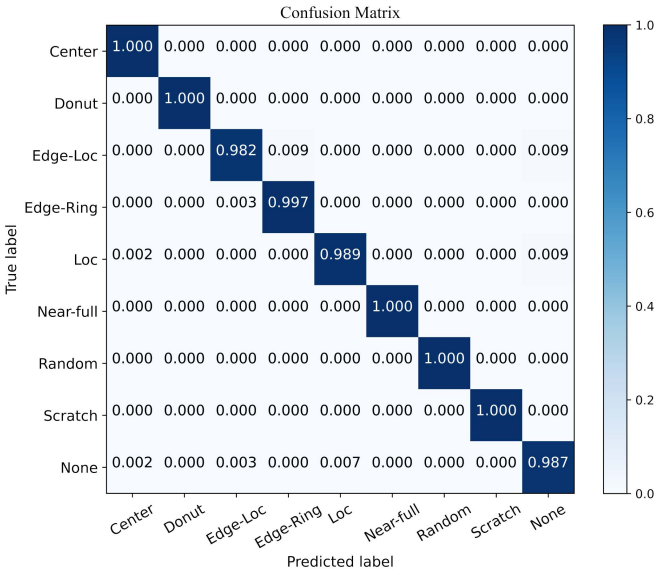

Fig. 4. Confusion matrix of VAEDLM evaluating nine types of wafer defect.

TABLE III
MODEL EVALUATION RESULT COMPARED WITH OTHER MODEL

Model	Testing Acc	Precision	Recall	F1-score
SCSDAE[4]	92.63			
WMFPR[5]	94.63	-	-	-
DTE-FPR[10]	90.50	-	-	-
AdaBalGAN[16]	97.50			
SVE[11]	95.8616	96.7344	96.9326	96.7124
CNN-WDI[15]	96.2	96.4	96.2	96.2
VAEDLM(ours)	**98.94**	**98.98**	**98.90**	**98.94**

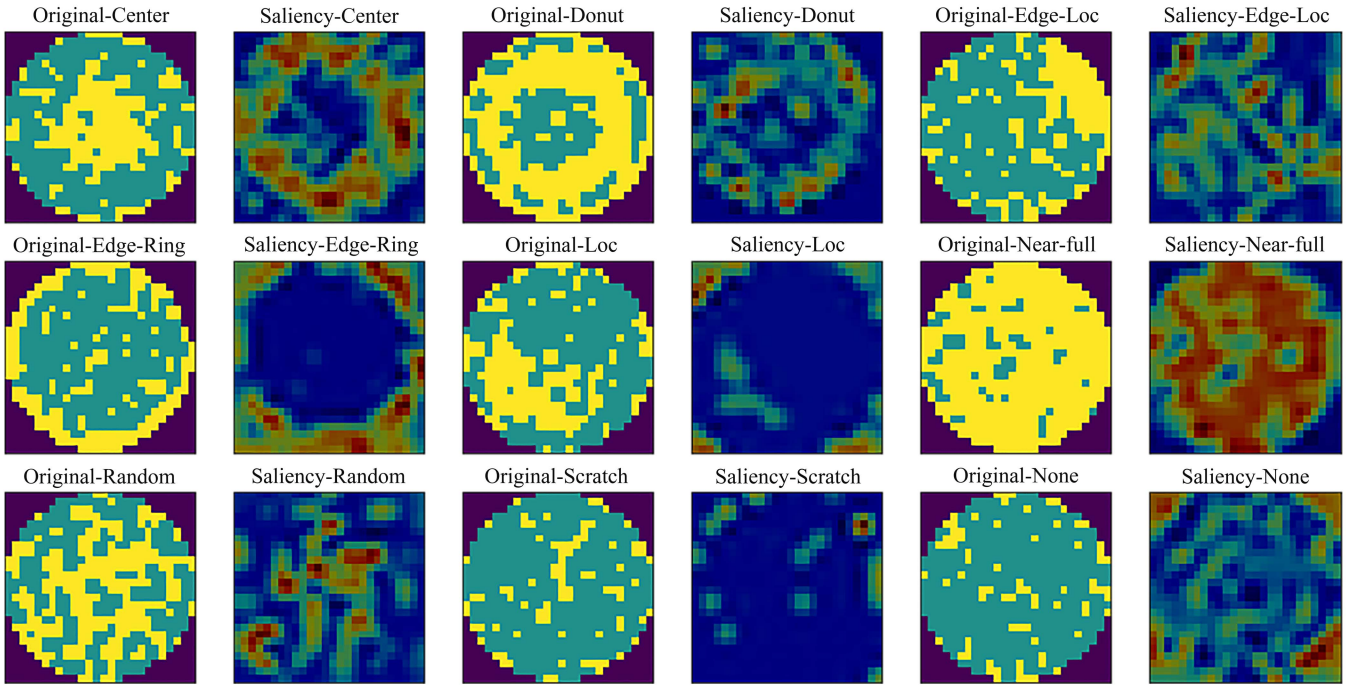

Fig. 5. Saliency map and its corresponding nine wafer defect images.

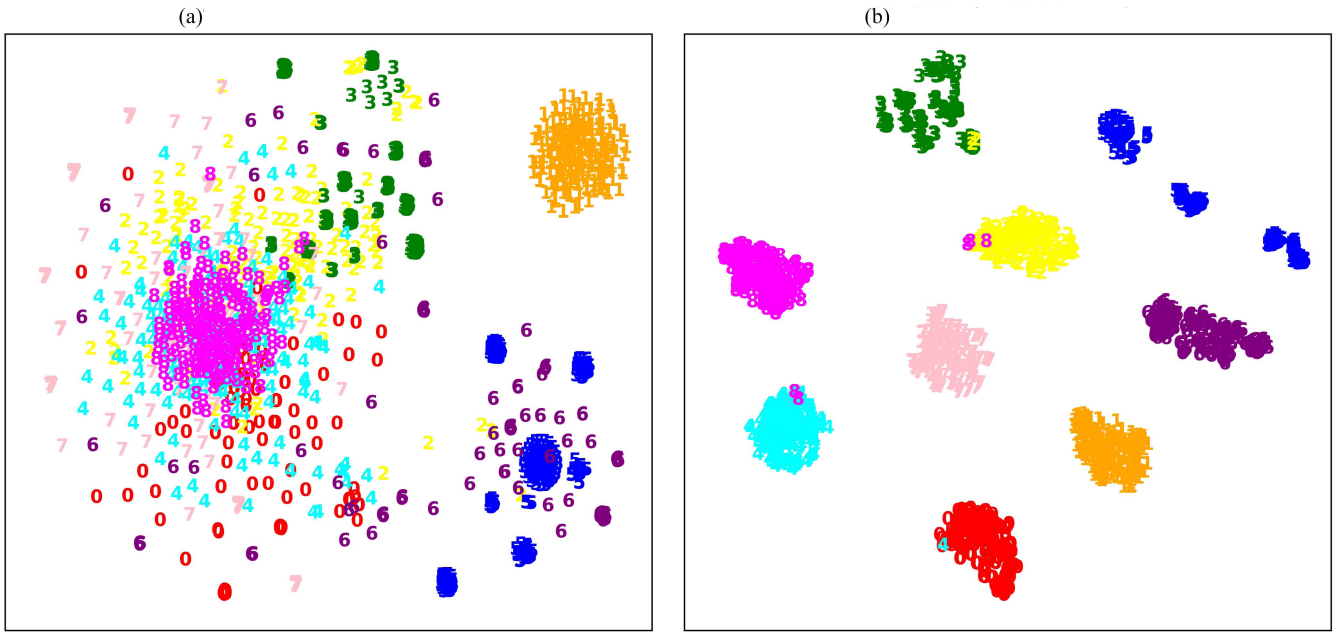

Fig. 6. t-SNE comparison before and after MLP processing. The numbers correspond to different types of defect. The relationships are follows: 0: Center, 1: Donut, 2: Edge-Loc, 3: Edge-Ring, 4: Loc, 5: Near-full, 6: Random, 7: Scratch, and 8: None. (a) t-SNE before MLP processing. (b) t-SNE after MLP processing.

CNN-WDI [15] is a deep learning-based model with several stacked convolution blocks. AdaBalGAN [16] is also a deep learning-based model with Generative Adversarial Network for data augmentation. It is evident the VAEDLM model outperforms the prior works in all of the metrics. It once again demonstrates that the proposed method is well performed on this task.

The saliency [20] maps can illustrate which part of the image the trained model considered critical to classify defects. It operates by computing the gradient values of the output class score with respect to input image pixel intensity. The higher the gradient value is, the pixel is more activated for classification. The saliency maps of the nine different defect images are shown in Fig. 5. This result indicates the neural network can managed to find the part of the defect occurrences. We noticed the parts where the model considered important for feature extraction are very similar to that of a human, meaning the network is indeed capturing the correct features.

To better understand and visualize the model's working process, we use t-SNE [21] to interpret the failure reasons. T-SNE can help visualize the high-dimensional data by mapping the clustered features into low-dimensional space. Nine different types of evenly distributed 1800 samples are used for demonstration. Integers between 0 and 8 are used to represent different defect types. As is shown in Fig. 6, after deep learning model's processing, the mapped feature points form denser clusters and become more separated from each other. We also notice the few cases that data does not cluster correctly. In Fig. 6(b), five points of "Loc" (8) cluster to "Edge-Loc" (2) or "Loc" (4), and three points of "Edge-Loc" (2) cluster to "Edge-Ring" (3). One point of "Loc" (4) clusters to "Center" (0). Comparing with the confusion matrix, we can find similar tendency. This might be resulted from the inherent similarities between these types of defects. "Near-full" (5) was divided into three clusters, which indicate the pattern type itself might be separable. The t-SNE study helped to refine the model and future work will be devoted to improve the model's performance on these similar defect types accordingly.

IV. Conclusion

Here, we proposed a deep learning-based method to automate wafer defect pattern recognition with high performance. It solved the wafer defect pattern imbalanced problem by augmenting the data with VAE. We used a fine-tuned deep CNN model to train the processed dataset and classify the defect. The overall method enhanced the averaged accuracy toward 99.19% after the data augmentation, which is a significant boost comparing to the accuracy without augmentation. It beats the state-of-the-art methods in accuracy, recall, precision, AUC, and F1-Score. The model is agile and has the potential to be applied in a semiconductor manufacturing foundry to cater for the high-throughput demand. Future works will be devoted to decrease misclassification between the similar patterns.

Code Availability

The code for this project is available at: https://github.com/shuyu-wang/wafer_defect_classification.

Competing Interests

The authors have declared no competing interests.

Authors Contributions

Shuyu Wang devised the project, the main conceptual ideas, and proof outline. He also wrote half of the manuscript. Zhitao Zhong performed the experiments, derived the models, and wrote the other half of the manuscript. Yuliang Zhao involved in planning the work and helped on the article editing. Lei Zuo contributed in manuscript improvement.

References

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

[2] T. Yuan and W. Kuo, "A model-based clustering approach to the recognition of the spatial defect patterns produced during semiconductor fabrication," *IIE Trans.*, vol. 40, no. 2, pp. 93–101, 2007.

[3] C.-W. Chang, T.-M. Chao, J.-T. Horng, C.-F. Lu, and R.-H. Yeh, "Development pattern recognition model for the classification of circuit probe wafer maps on semiconductors," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 2, no. 12, pp. 2089–2097, Dec. 2012.

[4] 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.

[5] 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.

[6] C.-J. Huang, C.-F. Wu, and C.-C. Wang, "Image processing techniques for wafer defect cluster identification," *IEEE Design Test Comput.*, vol. 19, no. 2, pp. 44–48, Apr. 2002.

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

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

[9] M. P.-L. Ooi, H. K. Sok, Y. C. Kuang, S. Demidenko, and C. Chan, "Defect cluster recognition system for fabricated semiconductor wafers," *Eng. Appl. Artif. Intell.*, vol. 26, no. 3, pp. 1029–1043, Mar. 2013.

[10] 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.

[11] 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, Mar. 2019.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] J. Wang, Z. Yang, J. Zhang, Q. Zhang, and W.-T.-K. Chien, "AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 3, pp. 310–319, Aug. 2019.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[18] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, Jun. 2013, pp. 1–6.

[19] J. Duchi, J. C. Bartlett, and P. L. Wainwright, "Adaptive subgradient methods for online learning and stochastic optimization," in *Proc. Conf. Decis. Control*, vol. 12, 2012, pp. 5442–5444.

[20] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998, doi: 10.1109/34.730558.

[21] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2625, Nov. 2008.