Journal of Intelligent Manufacturing (2023) 34:3215–3247
https://doi.org/10.1007/s10845-022-01994-1

Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: a review

Tongwha Kim[1] · Kamran Behdinan[1]

Received: 24 September 2021 / Accepted: 7 July 2022 / Published online: 23 August 2022

Abstract
With the high demand and sub-nanometer design for integrated circuits, surface defect complexity and frequency for semiconductor wafers have increased; subsequently emphasizing the need for highly accurate fault detection and root-cause analysis systems as manual defect diagnosis is more time-intensive, and expensive. As such, machine learning and deep learning methods have been integrated to automated inspection systems for wafer map defect recognition and classification to enhance performance, overall yield, and cost-efficiency. Concurrent with algorithm and hardware advances, in particular the onset of neural networks like the convolutional neural network, the literature for wafer map defect detection exploded with new developments to address the limitations of data preprocessing, feature representation and extraction, and model learning strategies. This article aims to provide a comprehensive review on the advancement of machine learning and deep learning applications for wafer map defect recognition and classification. The defect recognition and classification methods are introduced and analyzed for discussion on their respective advantages, limitations, and scalability. The future challenges and trends of wafer map detection research are also presented.

Keywords Wafer Map · Semiconductor manufacturing · Machine learning · Deep learning · Defect recognition · Defect classification

Introduction

Integrated circuits (IC) are the fundamental electronic component for many electronic devices and are developed on semiconductor wafer substrates. As the electronics industry demands for high levels of innovation, development, and competition (Ebayyeh & Mousavi, 2020), ICs are continuously developed and scaled to state-of-the-art design and complexity. With such growth, defect complexity and frequency have increased, which has subsequently warranted the greater need for accurate and real-time quality monitoring and control to promote high yield, cost-efficiency, and performance.

The onset of unknown/rare, mixed, and complicated defects ultimately results in increased costs, low product yield and deteriorated fabrication process stability. As such, utmost importance has been set upon defect detection and root-cause analysis (RCA) as defect patterns can indicate the potential causes of process variation. With the accuracy and time constraints of manual detection and the advances in algorithms, hardware, and data availability, machine learning (ML) and deep learning (DL) have been increasingly adapted and integrated into various domain applications (i.e., medical, manufacturing, finance), including surface defect detection for semiconductor wafer surfaces. During the wafer and IC fabrication processes, defects arise from process and equipment instability, as well as environmental factors such as airborne particles. Traditionally, wafer maps (WM)—which are visual representations of circuit probe [electrical] testing data—were used by engineers with high-level domain knowledge for manual defect recognition and classification. However, with the increase in design complexity and sub-nanometer IC design, automated detection and recognition is increasingly sought after (Liu & Chien, 2013).

✉ Tongwha Kim
tongwha.kim@mail.utoronto.ca

Kamran Behdinan
behdinan@mie.utoronto.ca

[1] Advanced Research Laboratory for Multifunctional Lightweight Structures (ARL-MLS), Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, ON, Canada

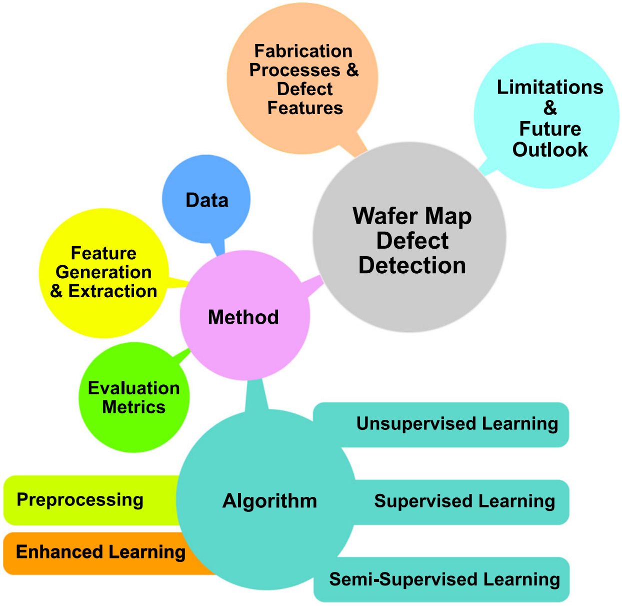


Fig. 1 Mind map illustrating overall structure of paper

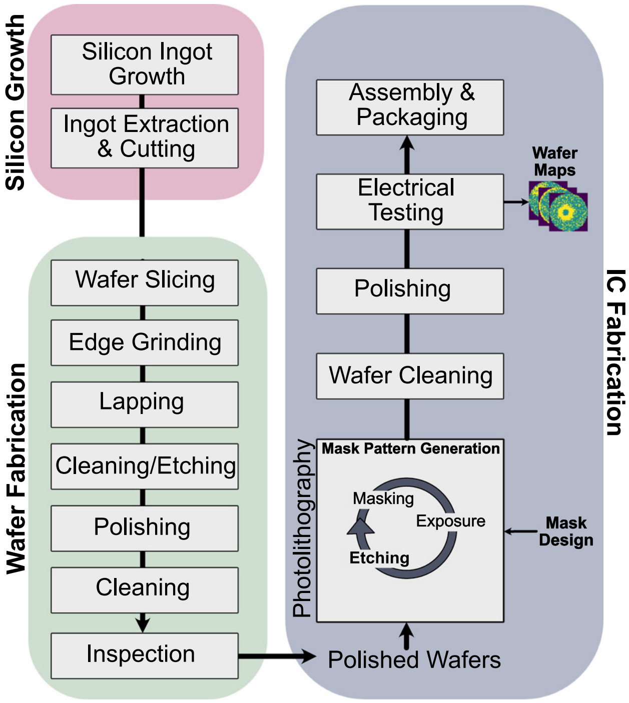


Fig. 2 General wafer and IC fabrication processes

With recent developments in computer vision and ML/DL techniques, defect recognition and classification algorithms were further enhanced, wherein the respective applications focused on improving overall performance, cost-efficiency, and runtime. Even reinforcement learning has been leveraged as a search algorithm for optimal parameters and architectures (Baker et al., 2017; Bello et al., 2017; Shon et al., 2021). Various model architectures, algorithms and learning mechanisms have been explored to achieve state-of-the-art performance. As such, this paper concentrates on the various ML and DL applications for WM defect recognition and classification. This paper introduces and compares the various wafer map defect detection algorithms, along with the discussion of their respective advantages, and limitations. The current challenges and future research trends of WM defect recognition and classification are also presented.

The rest of this paper is organized as follows: Section Background presents the background into fabrication processes, as well as the fundamental components for wafer map defect recognition and classification. Section Methodologies and learning strategies provides the details, analysis, and discussion of the advances of WM defect learning and detection algorithms. Finally, the conclusions are drawn in Section Discussion and conclusion, along with challenges and future trends. The mind map in Fig. 1 illustrates the structure of the paper.

Background

The current and future trends for wafer fabrication, specifically the evolving technologies and design standards, affect the production yield, defect complexity, and effectiveness of quality inspection technologies. Similarly, with the progression of ML, DL, and computer vision, the algorithms for wafer map defect detection (WMDD) have incorporated these methods to enhance model performance regarding accuracy, computational load, run-time, and learning capability. This section introduces the semiconductor wafer fabrication and inspection processes, as well as the fundamental components of the ML/DL applications for WM defect recognition and classification.

Semiconductor wafer fabrication and inspection

Semiconductor wafers are the silicon-based substrates used to fabricate ICs. The application and scale of ICs require precise manufacturing and strict quality control. The general wafer and IC fabrication line is shown in Fig. 2, including the quality inspection checkpoints. The major stages of wafer fabrication and inspection are briefly described below.

Wafer fabrication starts at silicon ingot growth and extraction. Mono-crystalline or poly-crystalline silicon is used for silicon growth. In practice, silicon ingots are typically grown using the Czochralski (CZ) method or alternatively, the float-zone (FZ) method (Airaksinen et al., 2015; Cuevas & Sinton, 2018). Note that the growth method may impact production costs, and material properties, such as thermal stress resistance. After ingots are grown, they are extracted and cropped to remove the non-cylindrical ends. The silicon ingot is then sliced into thin wafers by (diamond) wire cutting. For the purposes of wafer tracking, wafers are marked with characters to indicate manufacturing information (i.e., identification, dopants, orientation) (Airaksinen et al., 2015). Afterwards, using a profiled diamond wheel, the wafer edges are grinded to a standardized or customized edge profile to adjust diameter, and minimize risk of slipping and chipping (Airaksinen et al., 2015). Resulting from the prior cutting process, the wafer surface is susceptible to large total thickness variations (TTV), which disposes the surface to additional process variations from downstream processes. As such, lapping or single-sided grinding is conducted to achieve TTV, surface roughness, and thickness measures within acceptable standard ranges. Residual mechanical damage may develop on the surface and/or edges after the lapping and grinding operations (Airaksinen et al., 2015). To remove the damage and any remaining impurities, chemical etching (alkaline or acidic) is conducted. Subsequently, the wafers undergo polishing to achieve desired thickness, TTV, and flatness. Then the polished wafers undergo a cleaning sequence and quality inspection prior to IC fabrication. Quality inspections for wafers involve measuring the physical, material, and chemical properties of the finished product with respect to standard and design specifications (Airaksinen et al., 2015; Cuevas & Sinton, 2018). For surface inspections, wafer defect detection systems leverage WM images, or wafer images. WM images are the spatial results from electrical testing, which illustrate individual die functionality, such that defect patterns are clusters of faulty dies. Wafer bin maps (WBM) are the resulting binarized WM images. Wafer images are generated from automated visual or electron beam inspection systems (Patel et al., 2020). Automated visual inspection systems typically utilize optical imaging techniques, including scanning acoustic tomography (SAT) (Chen, 2020), scanning electron microscopy (SEM) (Kim & Oh, 2017; Cheon et al., 2019), and charged-coupled device (CCD)-based imaging (Chen et al., 2020a, 2020b; Wen et al., 2020).

IC fabrication consists of photolithography, assembly, and packaging. Photolithography is used to pattern the wafer, and involves a repetition of various steps: masking, exposure, and etching. Mask design is used to develop the desired patterns for masking; inverse-lithography technologies (ILT) determine the optimal mask to achieve the desired wafer patterns, and is emerging as a prominent research field (Shi et al., 2019, 2020). Masking involves the application of photoresist, and photomask alignment to the wafer. Then the wafer is exposed to ultraviolet (UV) light through the photomask to reveal the patterns, which is followed by etching. Using chemical processes, etching develops and removes the exposed photoresist and exposed oxide layer. To create the desired IC patterns, photolithography is repeated in cycles for pattern and structure development. After the dies (*also known as chips*) have been developed, wafers undergo a sorting test, which involves electrical testing to determine die functionality. As part of assembly and packaging, the wafer is sliced into individual pieces, in which the faulty dies are discarded, and the remaining dies are forwarded to packaging.

The current technologies and design standards for IC fabrication are evolving, specifically for photolithography and IC design. Current designs and lithography technologies are at the sub-10 nm scale, specifically with extreme ultraviolet (EUV) lithography (Hasan & Luo, 2018; Preil, 2016). With competition and fast-evolving technologies, the future trends for IC fabrication include sub-5 and sub-3 nm scale lithography. As these future trends and technologies are realized, defect frequency and complexity increase, subsequently increasing the emergence of unknown, rare, and mixed-type defects; rendering defect detection more difficult, and emphasizing the need for more robust and reliable detection methods. The wafer production and IC fabrication processes, associated defects and causes are summarized in Table 1.

Table 1 Summary of processes and associated defects

Defect	Associated process/stage	Cause
Random	Clean room	Environmental conditions of clean room may induce particles and debris onto wafer surface
Loc	Lapping, grinding Polishing	Non-uniform surface Uneven cleaning
Edge-Loc	Lapping, grinding Polishing	Non-uniform surface Uneven cleaning
Center	Polishing	Non-uniform surface during chemical mechanical process (CMP)
Edge-Ring	Lapping, grinding Photolithography	Non-uniform surface Layer-to-layer misalignment Chemical etching issues
Scratch	Assembly, packaging Polishing Clean room	Mishandling Hardening of polishing pads Agglomeration of particles
Near-full	–	Agglomeration of multiple systematic and random defects
Donut	Lapping, grinding Polishing	Non-uniform surface Equipment handling or hardening of polishing pads

Data

The data is the input used to train, validate, and test the models. They originate from real-world fabrication lots or are simulated via generative modeling. Generative models learn the probability distribution of the data, such that data can be generated by sampling from the learned distribution (Kingma et al., 2014; Ruthotto & Haber, 2021). With the advent of powerful, and deep generative models, many studies have applied generative modeling for the purpose of wafer map data generation (Ji & Lee, 2020; Lee & Kim, 2020; Wang et al., 2019). Wafer map data generation can also be used as a data augmentation tool to tackle class imbalance and is introduced later in-depth in Section Enhanced learning strategies.

Across the wafer map defect detection literature, two defect classes have been identified as random, and systematic. Random defects are caused by environmental factors within the manufacturing space, such as air particles and are globally distributed across the wafer surface. They have no identified association to fabrication processes, and as such are typically removed during image preprocessing. Systematic defects are caused by process deviations and have localized spatially correlated patterns. The root causes for systematic defects have been identified and associated to specific fabrication processes (Table 1).

Single-type defect wafer maps reflect the presence of a single defect pattern, in which labels indicate the most salient defect pattern. Mixed-type defects are the agglomeration of random defects and two or more systematic defect patterns. It is important to note that majority of past works have focused on the detection of single-type systematic defects. However, with the onset of complex defects, recent works have shifted focus onto mixed-type defect recognition and classification. Shown in Fig. 3 are various examples of normal, single-type and mixed-type defects from the Mixed WM-38 dataset (Wang et al., 2020). Wafer maps labeled as *normal* are without defects.

For data sourcing, there are publicly available and private datasets. The WM-811K data (Wu et al., 2015) is a prominently used public dataset and is heavily featured in past works. The WM-811K dataset consists of 811,457 wafer bin maps from 46,393 real-world fabrication lots, and other manufacturing process data, including die size, lot name and wafer index. The labels are single-type defects; however, it is important to note that the labels reflect the most salient defect pattern, despite the presence of mixed-type defects. The exploratory data analysis for this dataset (Fig. 4, top) revealed majority of the data is unlabeled, and amongst the labeled wafer maps, a majority are labeled as *Normal*. Another publicly available dataset is the Mixed WM-38 (Wang et al., 2020). The Mixed WM-38 dataset consists of 38,015 wafer bin maps of mixed-type defects. This dataset includes 38 defect classes, which consists of 29 mixed-type defects of 2-, 3-, and 4-mixed types, 8 single-type defects, and normal (non-defect). Exploratory data analysis (Fig. 4, center) has shown that most mixed-type wafer maps have two to three defect types. In Fig. 4 (bottom), the distribution of defect classes with respect to defect types is shown. Pleschberger et al. (2019) collected a total of 1000 wafer maps from five lots, which included five different classes of simulated defect patterns with varying degrees of Gaussian noise. Each WM is described to contain approximately 17,000 devices (dies), in which the (*x,* y) spatial coordinates are given, along with their respective electrical testing results. Beyond the public datasets, many developments obtained private WBM datasets directly from semiconductor manufacturing companies (Adly et al., 2015b; Bella et al., 2019; Hwang & Kim, 2020; Tello et al., 2018). As wafer map labelling is manually conducted by domain engineers, which is time consuming and expensive, majority of the provided data were limited in size and types of defects. Note that wafer map datasets typically have two limitations: (1) severe class imbalance, and (2) lack of labels. Class imbalance is the unequal proportions of data examples for each class. Severe class imbalance persists in the datasets as wafer defects appear at lower frequencies than normal wafer maps. As such, datasets typically lack labels and have an abundance of unlabelled wafer maps. It should be noted that with manual annotation, there exists incorrect and/or uncertain labelling due to human error (Northcutt et al., 2021; Park et al., 2020).

Features

Features capture the intrinsic information from the input data and are a critical component as feature learning can bottleneck model performance. Respective to the model and learning strategy, features are derived from feature generation or feature extraction.

Feature generation is the process in which features are engineered from raw data transformations. Prior to the onset of neural networks, past works relied on manual feature generation for effective features as input data to classifiers (Mohanaiah et al., 2013; Ooi et al., 2013; Saqlain et al., 2019; White et al., 2008; Wu et al., 2015; Yu & Lu, 2016). These past works have included generated features such as: (a) *geometrical features*, (b) *Radon projection features*, (c) *density features,* (d) *texture features*, and (e) *gray features*, which are described in Table 2. Through manual feature generation, original features are obtained and used for model training. The main advantages of manual feature generation are that these features require minimal storage and computation, and that domain knowledge can be instilled during feature engineering, which can be especially beneficial for well-known and heavily studied defect patterns (Saqlain et al., 2019).

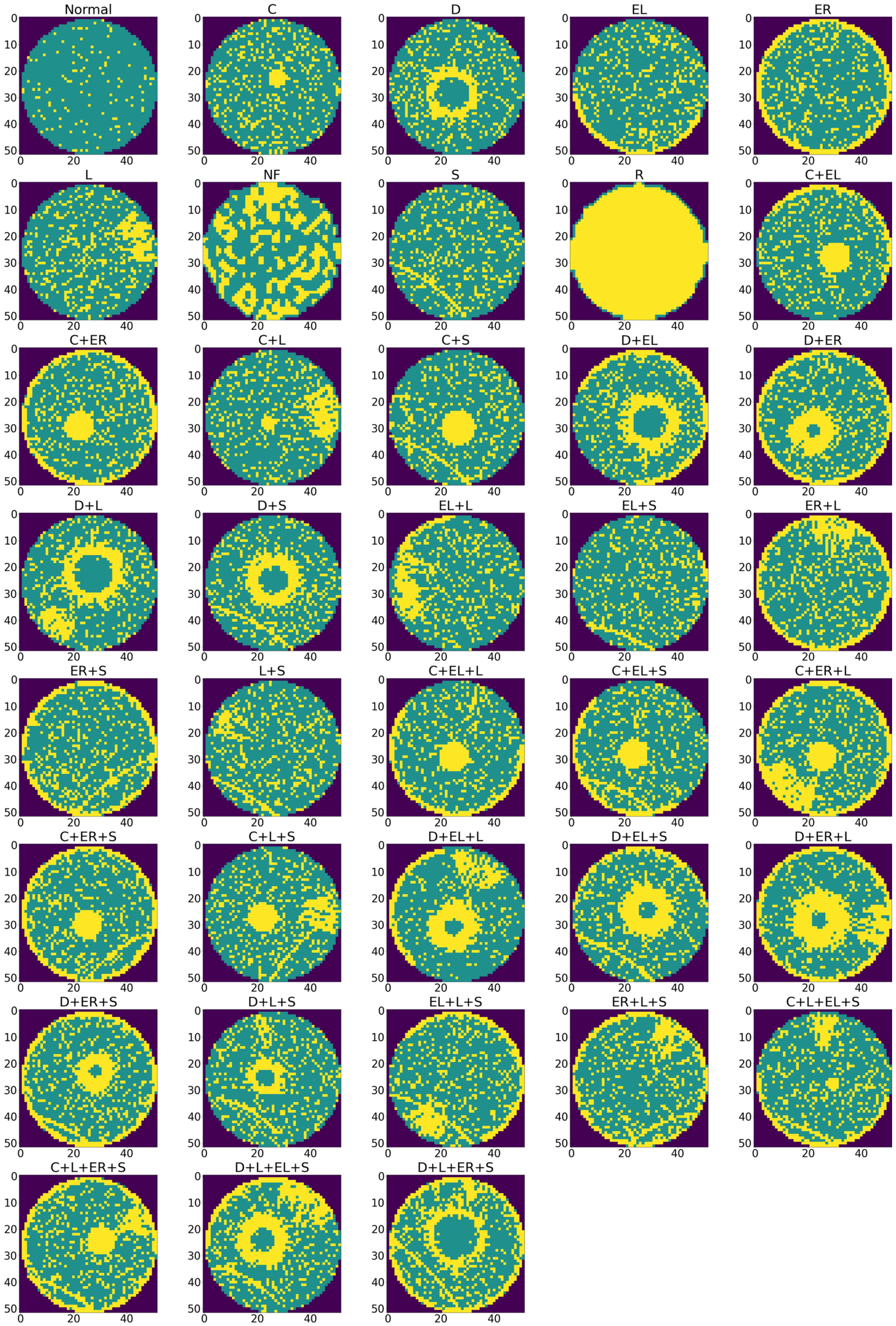


Fig. 3 Normal, single-type, and mixed-type defects with image dimensions of (52, 52) from Wang et al. (2020)

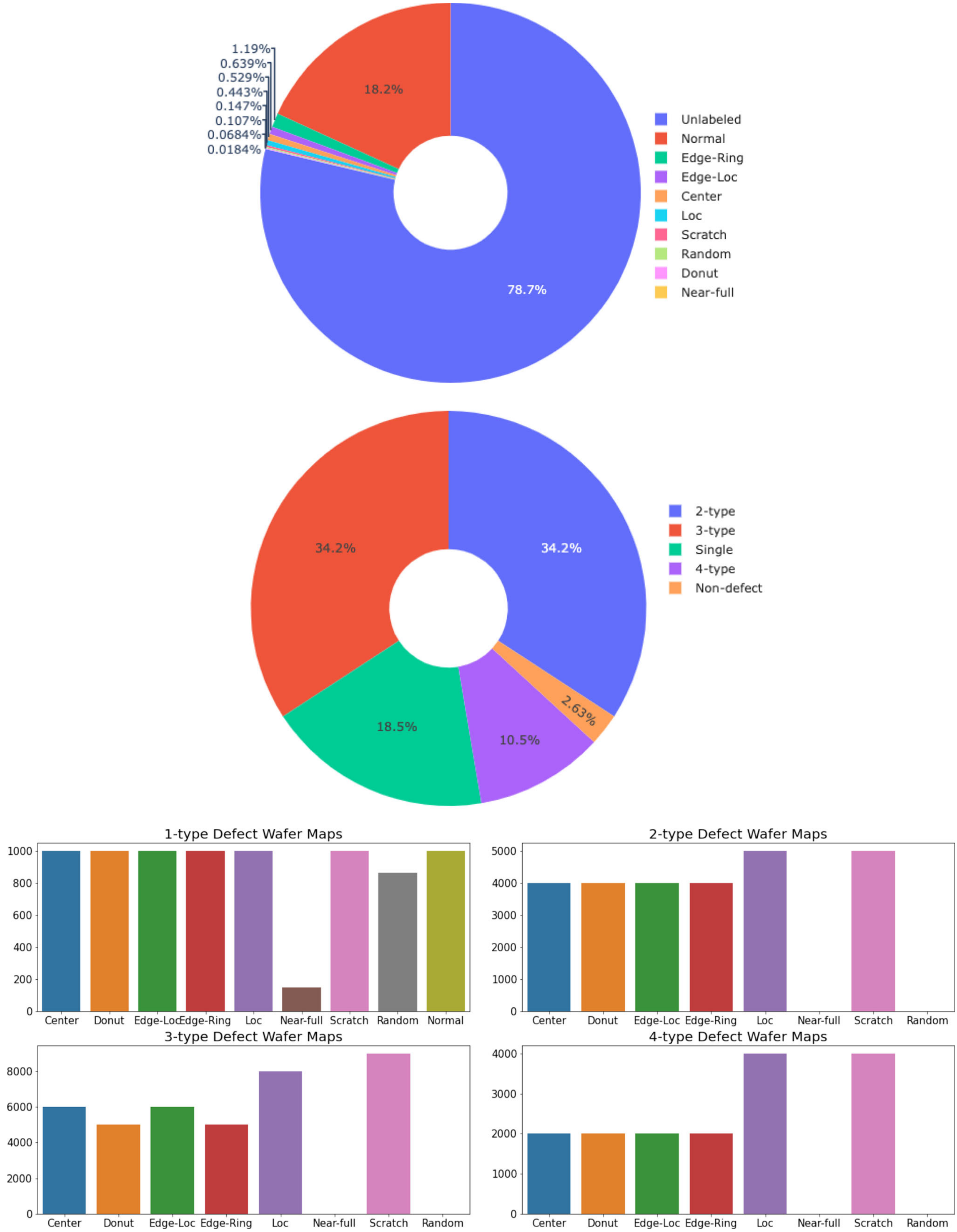


Fig. 4 Defect class distribution for WM-811 K (top) and Mixed WM-38 (center, bottom) datasets

Table 2 Summary of generated feature types and usage in literature

Class	Papers	Feature	Description
Geometrical	White et al. (2008) Chang et al. (2012) Wu et al. (2015) Fan et al. (2016) Wang and Ni (2019) Saqlain et al. (2019) Kang and Kang (2021)	Area	Reflects the area of the defect pattern, typically the most salient region (single-type) is considered. Also expressed as ratio of defect pattern area to wafer map area
		Perimeter	Defines the perimeter of the defect pattern. Also expressed as ratio of defect pattern perimeter to radius of wafer map
		Convexity	Indicates the convexity of a defect pattern, and is expressed as the ratio of the convex hull's perimeter to the total perimeter of the defect
		Length of Major/Minor Axes	Computes the length of the major/minor axes of the approximated ellipse that surrounds the defect pattern (most salient region)
		Eccentricity	Describes the outline shape of the approximated ellipse that surrounds the defect pattern
		Solidity	Estimates the proportion of defective die in the convex hull of the defect pattern
		Hough Transform	Identifies edges and lines in defect patterns
		Hu Invariant Moments	Set of seven values for central image moments. Image moments can recognize patterns independent of size, position, and orientation (Hu, 1962)
Projection	Wu et al. (2015) Yu and Lu (2016) Piao et al. (2018) Saqlain et al. (2019) Kang and Kang (2021)	Radon Transform	Image projections at various angles are collected as a 2D representation of the wafer map. Projections obtain geometric/structural information specific to defect patterns
Density	Fan et al. (2016) Saqlain et al. (2019) Kang and Kang (2021)	–	Reflects the computed failed die density distribution. Involves dividing the wafer map into multiple segments, and computing the defective die density per segment
Texture	Yu and Lu (2016)	–	Describes and extracts surface textural features from images using the statistical method, gray level co-occurrence matrix (GLCM). Examples of textural features are correlation, entropy, energy, contract, and uniformity (Mohanaiah et al., 2013)
Gray	Yu and Lu (2016)	–	Reflects the pixel distribution in images. Is typically expressed with various statistical features, including mean, variance, skewness, and kurtosis

However, this advantage also poses as a caveat to generating effective, handcrafted features because the degree of domain knowledge may not be sufficient to represent and differentiate the different defect patterns (Kang & Kang, 2021; Yu & Lu, 2016). This also imposes a limitation in detecting rare/unknown defects in regards to forming features: important characteristics of these defects may not be known or understood to generate effective features for detection and classification.

In contrast to feature generation, feature extraction can be applied to raw data, such as the wafer map images. Feature extraction includes dimensionality reduction techniques, and representation learning. Dimensionality reduction techniques, like principal component analysis (PCA) and linear discriminant analysis (LDA) are applied to extract the critical features for a lower dimensional representation (Wang & Ni, 2019; Yu & Liu, 2020). As information is lost when transforming into a lower dimensional space, PCA aims to minimize the number of features while maximizing the amount of variance captured by set of features. A major limitation of PCA is that it does not consider spatial relations within the data, such that the underlying patterns are not effectively captured. On the other hand, LDA weakly maintains spatial relations by using class labels to instill low-level discriminatory power in separating classes in the lower dimensional subspace (Wang et al., 2014). Despite these limitations, dimensionality reduction techniques can reduce computational complexity, and improve model performance. Additionally, research into non-linear dimensionality reduction (manifold learning) techniques have demonstrated improved retention of spatial relations, including autoencoders (AE), t-distributed stochastic neighbor embedding (t-SNE), locally linear embedding (LLE), multi-dimensional scaling (MDS), and isomap (Faaeq et al., 2018).

Representation learning is automated feature extraction. Deep learning models, like convolutional neural network (CNN)—which are neural networks that employ nonlinear kernels for learning shared weights for input feature maps, have been highly used in various computer vision tasks due to the automated feature extraction ability (Nakazawa & Kulkarni, 2018; Park et al., 2020; Shen & Yu, 2019). The automated feature extraction learns rich and highly descriptive features at each convolution layer. Similarly, representation learning can also be conducted via inference models. Probabilistic generative models, such as variational autoencoders (VAE) and generative adversarial networks (GAN), leverage inference methods to approximate and learn latent feature representations of the data via latent variable(s) z (Kingma et al., 2014; Kong & Ni, 2020a). It is important to note that the latent space embeds the input to a compact, and non-linear representation. Depending on the learning approach, automated feature learning can be executed with labeled and/or unlabeled data. With representation learning, raw data can be used, and can gain high discriminatory power as the underlying structure of the data can be learned, demonstrating capability with complex patterns and data structures (Khastavaneh & Ebrahimpour-Komleh, 2020; Zhong et al., 2016). The significance of representation learning is demonstrated with transfer learning (Section Enhanced learning strategies), wherein the feature extractor networks (backbone) of pretrained models have gained strong feature extraction capabilities to extract meaningful features (Chien et al., 2020; Ishida et al., 2019; Shen & Yu, 2019). However, the capacity of representation learning is constrained by model complexity, as performance is dependent on whether the model is suited to the respective data complexity and problem.

With the onset of neural networks, research has shifted from manual feature generation to feature representation learning as leveraging feature learning algorithms has proven to generate more meaningful and effective features for downstream tasks, especially for problems with complex data structures.

Algorithms for wafer map defect detection

The algorithms are the learning strategies in which the model learns and trains from the input data. In this section, the three learning strategies that we will focus on are introduced: supervised, unsupervised, and semi-supervised learning. The main algorithms for wafer map defect detection are discussed in-depth in Section Methodologies and learning strategies. In Table 3, the prominent works for each main algorithm used in wafer map defect detection are listed.

Table 3 Selection of prominent machine learning and deep learning algorithms for wafer map defect detection

Algorithm	Approach	Paper
Supervised	Conventional ML Classifiers (i.e., SVM, decision trees, ensembles)	Piao et al. (2018) Saqlain et al. (2019) Kang and Kang (2021)
	Neural Networks	Kyeong and Kim (2018) Nakazawa and Kulkarni (2018) Kim et al. (2021) Wang et al. (2020)
Unsupervised	Mixture Models	Kim et al., (2018) Ezzat et al. (2021)
	Density-based (DBSCAN, OPTICS)	Jin et al. (2018)
Semi-supervised	Pretraining-Finetuning	Yu (2019) Shon et al. (2021)
	Generative Modelling	Kong and Ni (2018) Hu et al. (2021) Yu et al. (2019b) Yu and Liu (2020) Lee and Kim (2020) Kong and Ni (2020a)
Enhanced Learning	Model Optimization	Bello et al. (2017) Jang et al. (2020) Shon et al. (2021)
	Incremental Learning	Shim et al. (2020) Kong and Ni (2020a)
	Data Augmentation	Wang et al. (2019) Saqlain et al. (2020)
	Transfer Learning	Shen & Yu (2019) Ishida et al. (2019) Chien et al. (2020)

Supervised learning utilizes labels for model training, and loss functions, which measure the error between the predictions and ground truth. The labels are factored into the loss function, and acts as the supervisory signal for the model to learn the mapping for an input and the respective desired model output. Loss functions are optimized by finding the global minimum or optimal local minimum. It should be noted that loss functions are dependent on the downstream task, and their mathematical optimization is constrained by the convexity. The problem for supervised wafer map defect detection is defined as classification, in which the algorithms aim to learn the mapping from input to output to predict specific defect patterns. Early literature has transitioned from conventional machine learning classifiers to neural networks. Conventional machine learning classifiers typically require extensive preprocessing and manual feature generation and have mainly been applied for single-type defect detection. Common classifiers used in WM defect detection include SVM, decision trees, and ensembles (Fan et al., 2016; Piao et al., 2018; Saqlain et al., 2019; Wu et al., 2015). Neural networks are prominently used throughout the literature and have demonstrated capability for single-type and mixed-type defect classification.

It is important to note that the classification problem can be multi-class or multi-label. In multi-class classification, there are a distinct number of classes that the classifier learns and models. Each data sample belongs to a single class, and the classifier predicts the probability across all classes that the data sample belongs to a particular class. Multi-label classification is a multi-output algorithm, such that the data examples can be annotated with multiple target classes. For multi-class neural networks, the softmax function is used in the final output layer to compute the decimal probabilities, which add up to 1.0. On the other hand, multi-label neural networks utilize the sigmoid function in the final output layer to predict the probabilities (between 0 and 1) for each class. Mixed-type defect detection can be framed as a multi-class or multi-label classification problem. As a multi-class classification problem, mixed-type defects are segmented into multiple single-type defect patterns and are subsequently classified with a network of binary classifiers (Kong & Ni, 2019, 2020b; Kyeong & Kim, 2018). On the other hand, as a multi-label classification problem, mixed-type defect detection aims to recognize the different patterns and predicts the probability per class label for a single wafer map (Lee & Kim, 2020; Wang et al., 2020).

Unsupervised learning algorithms leverage unlabeled data to learn their underlying patterns, and structure. For wafer map defect detection applications, the main unsupervised learning tasks are clustering, and pretraining. Clustering focuses on self-organization to cluster data based on similarity and dissimilarity distances. Popular clustering algorithms for wafer map defect detection include density-based spatial clustering of applications with noise (DBSCAN), ordering point to identify the cluster structure (OPTICS), and mixture models, such as Gaussian mixture models (GMM) and infinite warped mixture models (iWMM) (Ezzat et al., 2021; Fan et al., 2016; Iwata et al., 2013; Kim et al., 2018). Spatial clustering applications in WMDD aim to segment the different defect patterns for both single-type and mixed-type defects. Unsupervised methods have also been leveraged for pretraining to supplement supervised methods with unsupervised feature representation learning using autoencoders (Shon et al., 2021; Yu, 2019). By taking advantage of the plethora of unlabeled data, unsupervised pretraining methods operate to learn general feature representations to better initialize the model weights for supervised training (relative to zero or random initialization) via reconstruction errors.

Semi-supervised learning leverages both labeled and unlabeled data for the model training process. During the training process, the labeled data is utilized in the same manner as supervised learning, whereas the unlabeled data is leveraged for transduction-based inference learning. This is reflected in the loss function, where a combined, and weighted loss function is defined to account for both labeled and unlabeled data. With transduction-based inference learning, all available data is observed to enhance the learned data representations for inferring missing labels. Relative to the former learning strategies, development of semi-supervised algorithms is growing to overcome the limitations imposed by supervised and unsupervised learning. For WMDD, pretraining-finetuning and semi-supervised generative models have been implemented to tackle the real-world issue of limited annotated wafer maps. For pretraining-finetuning, unsupervised pretraining methods are followed by supervised finetuning. Semi-supervised generative models are probabilistic methods, which include the models: variational autoencoders (VAE), and modified Ladder networks (Kong & Ni, 2020a; Lee & Kim, 2020). These methods have been applied towards both single-type and mixed-type defect patterns.

Beyond the model training algorithms, enhanced learning algorithms and techniques have been applied for wafer map defect detection to boost performance, and to address the issues with labeled data availability, class imbalance, rare/unknown defect detection, and model sensitivity. These algorithms and techniques have been introduced as data augmentation, incremental learning, transfer learning, and model optimization (Bello et al., 2017; Jang et al., 2020; Ji & Lee, 2020; Shim et al., 2020).

Evaluation

Evaluation methods are used to assess the performance, and can be conducted at validation, or the final testing stage. The results from the validation stage drive hyperparameter tuning and model optimization. The evaluation methods are dependent on the data and learning approach. Across the wafer map defect detection literature, the common performance evaluation indices have been identified and summarized in Table 4 below (Hwang & Kim, 2020; Kim et al., 2018; Lee &

Table 4 Summary of common performance evaluation indices

Learning approach	Performance Evaluation Index	Equation	
Supervised, Semi-supervised	Accuracy (Top-1)	$Acc = \frac{TP+TN}{TP+FP+TN+FN}$	(1)
	Precision	$Pre = \frac{TP}{TP+FP}$	(2)
		$Pre_i = \frac{TP_i}{TP_i+FP_i}$	(3)
	Recall	$Re = \frac{TP}{TP+FN}$	(4)
		$Re_i = \frac{TP_i}{TP_i+FN_i}$	(5)
	F-Measure	$F_1 = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Re}call}{\mathrm{Pr}\,ecision + \mathrm{Re}call}$	(6)
	Micro-precision	$MPre = \frac{\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j \hat{y}_i^j}{\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j} = \frac{\sum_j TP^j}{\sum_j TP^j + \sum_j FP^j}$	(7)
	Micro-recall	$MRe = \frac{\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j \hat{y}_i^j}{\sum_{i=1}^{N}\sum_{j=1}^{C} \hat{y}_i^j} = \frac{\sum_j TP^j}{\sum_j TP^j + \sum_j FN^j}$	(8)
	Exact match ratio	$EMR = \frac{1}{N}\sum_{i=1}^{N} I(y_i = \hat{y}_i)$	(9)
	Hamming loss	$Hamming\ loss = \frac{1}{NC}\sum_{i=1}^{N}\sum_{j=1}^{C} I(y_i \neq \hat{y}_i^j)$	(10)
Unsupervised	Rand Index	$RI = \frac{a+b}{\binom{n}{2}} = \frac{TP+TN}{TP+FP+FN+TN}$	(11)
	Adjusted Rand Index	$ARI = \frac{RI - \mathrm{E}[RI]}{1 - \mathrm{E}[RI]}$	(12)
	Normalized mutual information	$NMI = \frac{I(\mathrm{X};\mathrm{Y})}{\sqrt{\mathrm{H(X)H(Y)}}}$	(13)
	Adjusted mutual information	$AMI = \frac{MI - \mathrm{E}[MI]}{\max(H(U), H(V)) - \mathrm{E}[MI]}$	(14)
	Purity	$Purity = \frac{1}{N}\sum_{i=1}^{k} \max_j \lvert c_i \cap t_j \rvert$	(15)

Kim, 2020; Li et al., 2021; Saqlain et al., 2019). In Table 4, the variables *TP, TN, FP,* and *FN* represent True Positives, True Negatives, False Positives, and False Negatives respectively.

The evaluation methods for supervised learning algorithms indicate how well the model has learned by the number of correct and incorrect predictions. The (top-1) accuracy, precision, recall, and confusion matrix are the metrics typically used to evaluate and compare models. Equations (1) to (5) represent the (top-1) accuracy, precision, and recall. The accuracy indicates the total number of correctly identified wafer maps; precision signifies the total correctly identified wafer maps from all identified wafer maps, and recall indicates the total number of correctly identified wafer maps within a given set. Note that Eqs. (2) and (3), as well as Eqs. (4) and (5) represent the same equation, but are qualified by the given class *i,* such that the precision and recall are computed for each respective class *i*. The F-1 metric is the weighted average of precision and recall (Eq. 6), such that its respective score indicates how close the predicted and ground truth values are. These metrics are used to evaluate the multi-class classification performance for wafer map defect detection.

In the context of multi-label classification problems, exact match ratio (EMR), micro-precision (MPre), micro-recall (MRe), and Hamming loss can be used (Lee & Kim, 2020; Santos & Canuto, 2012; Wang et al., 2020). MPre (Eq. 7) and MRe (Eq. 8) differ from their multi-class counterpart by considering partially correct predictions, as each correct target label is counted for each sample $i \in N$, where N represents the total number of samples, and each class $j \in C$, where C represents the total number of known classes labels. On the other hand, EMR (Eq. (9)) is computed similarly to accuracy and reflects all fully correct predictions. Note that y_i and $\widehat{y}_i$ represent the true labels and predicted labels respectively, whereas ${y_i}^j$ and $\widehat{y}_i^j$ are the per class label equivalents. Hamming loss reflects the proportion of incorrectly predicted labels to the total number of labels at the individual label-level. As shown in Eq. (10), the indicator function evaluates to 1 when predicted labels do not match the ground truth label. Like MPre and MRe, N and C represent the total number of samples and total number of known classes respectively.

As semi-supervised algorithms leverage unlabeled data for label imputation, and feature representation learning during training, the performance is evaluated like supervised algorithms. Evaluation methods like accuracy, EMR, etc. are calculated on the labeled data.

For unsupervised wafer map defect detection methods, the performance indices typically evaluate the defect clustering results. The following have been identified and described by Eqs. (11) to (15) as the commonly used evaluation metrics for unsupervised defect detection algorithms: (i) Rand Index (RI), (ii) adjusted Rand Index (ARI), (iii) normalized mutual information (NMI), (iv) adjusted mutual information (AMI), and (v) Purity. These metrics focus on comparing the clusters via similarity, and shared information.

RI is the ratio of the number of correct similar pairs (a), and correct dissimilar pairs (b) to all possible combination pairs, where n represents the number of samples. ARI is the RI, but adjusted, such that independent of the number of clusters and samples, randomly clustered samples are closer to 0, and highly similar samples are closer to 1. In Eq. (12), $\mathbb{E}[RI]$ indicates the expected RI value. NMI (Eq. 13) is the normalization of mutual information (MI), which results in scores between 0 and 1. For Eq. (13), I(X; Y), H(X) and H(Y) represent the mutual information between X and Y, and the entropy of X and Y respectively. AMI is mutual information adjusted, such that permutations of the class and cluster labels would not affect the score. Lastly, purity (Eq. (15)) measures the accuracy of cluster assignments by tallying the number of correctly assigned samples and dividing by the total number of samples (N).

Methodologies and learning strategies

In this section, the recent developments in AI applications for WM defect recognition and classification are introduced, analyzed, and discussed. This section is organized into (1) preprocessing, (2) supervised learning, (3) unsupervised learning, (4) semi-supervised learning, and (5) enhanced learning strategies.

Preprocessing

The purpose of the data preprocessing stage is to preprocess and prepare the wafer map images for feature extraction and model training. Data preprocessing typically includes a multitude of operations for image transformations, and spatial filtering. Preprocessing operations include image size standardization, binarization, and denoising.

Image size standardization is to reshape the raw wafer maps to a single, uniform size, and utilizes interpolation algorithms to minimize quality loss. Interpolation algorithms are subject to the pixel neighborhood size for approximation, such that with increasing sizes results in longer rendering times and higher quality. The bicubic interpolation algorithm is typically used due to the optimal quality and time tradeoff. Binarization is used to convert wafer maps to wafer bin maps, in which individual die functionality is indicated by 0 s and 1 s.

Image denoising (outlier detection) and filtering refers to the process of removing random defects. It is typically conducted to enhance model performance and accuracy as the removal of random defects enhances the systematic defects. Past works have utilized spatial filtering and clustering methods to remove noise and isolate the systematic defects (Chien et al., 2013; Liu & Chien, 2013; Wang, 2008, 2009; Yuan et al., 2010). Spatial filtering algorithms focus on how to effectively differentiate between the random defects and the dies that belong to systematic defects. Spatial clustering algorithms focus on forming a separate cluster for each different defect patterns. The input to these methods has already filtered the defect patterns. Support vector clustering (SVC) has been used in Wang (2009) and Yuan et al. (2010) for defect denoising, and identification of systematic defect patterns. SVC demonstrated robustness against noisy data, but high sensitivity to defect complexity as clustering efficiency decreases with more complex defect patterns (i.e., multiple defects). Similarly, the k-nearest neighbors (kNN) algorithm is also used to differentiate between defective dies that belong to systematic defect patterns (Huang, 2007). The spatial randomness filter is a statistical method that checks the spatial independence of adjacent dies. The spatial independence is computed by taking the logarithm (Log) of the odds ratio ($\hat{\theta}$), in which the resulting Log$\hat{\theta}$ determines whether the wafer map is spatially random, contains a defect cluster, or repeating patterns (Chien et al., 2013; Liu & Chien, 2013). Although the filtering results indicate which wafer maps should be used for classification, as the spatial independence test is computed for the dies and not the pattern, misclassification can occur. Median filtering is a popular denoising method that replaces each die's value with the median value of the neighboring dies, and has been used in many works for image preprocessing (Kong & Ni, 2020a; Wang et al., 2006; Yu & Lu, 2016; Yu, 2019). Median filtering can be effective in removing the random defects, however, may also remove important pattern information as some of the systematic pattern dies may be removed. The thin geometries of the *Scratch,* and *Edge-Ring* defects are particularly sensitive to median filtering (Fig. 5). It is important to note that poor spatial filtering and spatial clustering can significantly affect

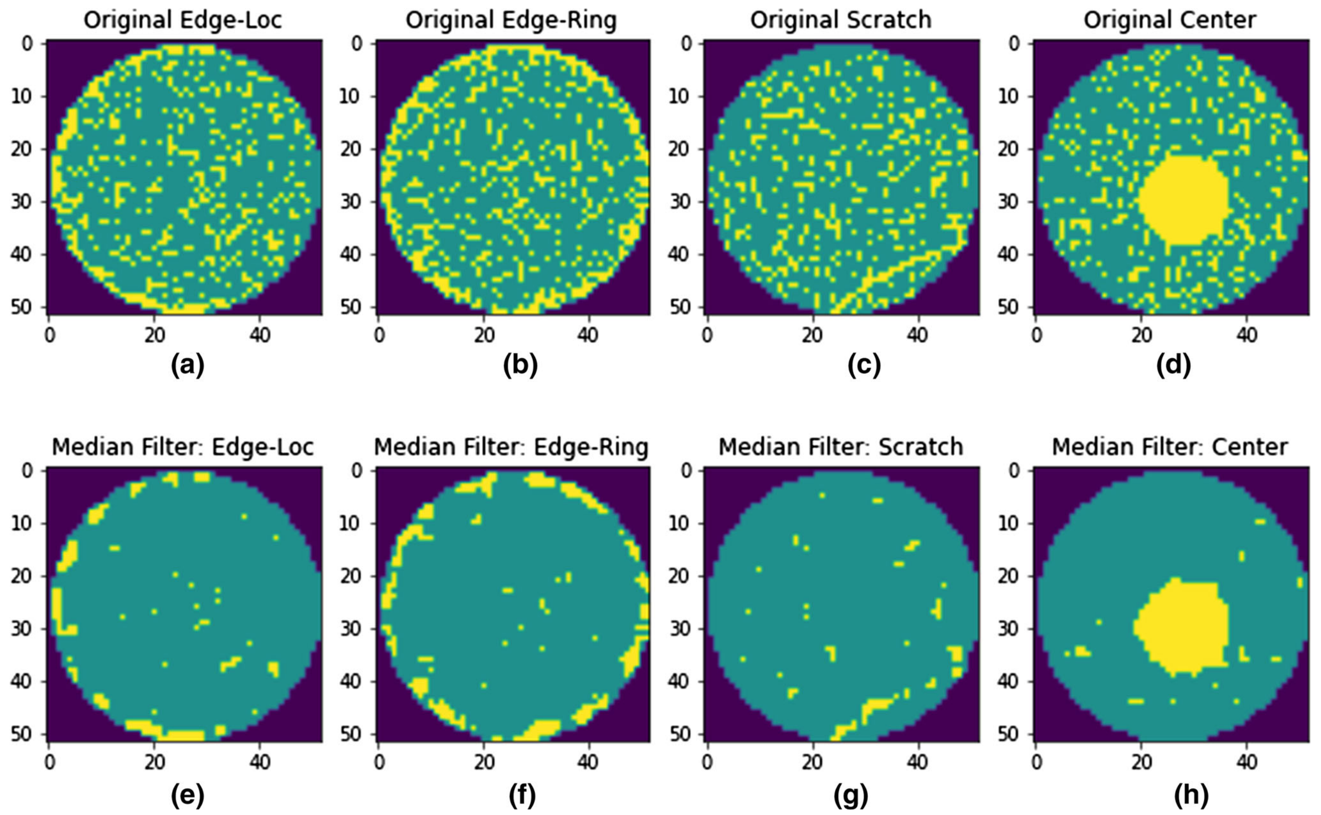


Fig. 5 **a**–**d** Original wafer maps, **e**–**h** Wafer maps after median filtering

downstream tasks as the filtered systematic defect pattern quality is damaged.

Wang and Chen (2019) proposed using three masking filters to preprocess wafer maps and extract rotation-invariant features for defect pattern classification. To address the limits of traditional spatial filtering methods for curvilinear and edge patterns, polar, line, and arc masks were applied at various angles to real-world wafer maps to extract features of concentric, linear, and eccentric patterns. Used to train various classifiers (i.e., neural networks, random forest, SVM), the masking filters demonstrated effectiveness with high defect recognition rates, but limited recognition for defect patterns with complex geometry (i.e., *Scratch*, *Reticle*).

The king-move neighborhood (Chien et al., 2013; Hsu et al., 2020; Wang, 2008; Wang & Ni, 2019), and Moore neighborhood (Jin et al., 2019) are utilized to compute the spatial correlation weights for the adjacent dies. Although both the king-move neighborhood and Moore neighborhood filters consider the eight surrounding dies, the Moore neighborhood filter also considers the center die. Typically, a global threshold criterion is applied to the spatial correlation weights, such that dies are removed if the criterion is not met. The downfall of using a global threshold criterion is that it does not consider the geometries and typical defect die densities of each defect, in particular the *Scratch* and *Edge-Ring* defects. According to Jin et al. (2019), their proposed DBSCAN-based algorithm considers defect pattern type for outlier detection. The outliers are completely removed for most defects (i.e., *Loc, Donut, Random*), and either carefully removed for the *Scratch* and *Edge-Ring* defects. The authors recommended to not completely remove the outliers for the *Scratch* and *Edge-Ring* defects as defect pattern quality would deteriorate.

The above filtering methods have demonstrated limitations towards the *Scratch* and *Edge-Ring* defects due to their thin and elongated shapes. As such, Kim et al. (2018) proposed the connected-path filtering (CPF) algorithm. The CPF algorithm uses depth-first search (DFS) to explore all possible paths between two defective dies, and recognizes the connected paths that are longer than a threshold criterion to represent the identified defective die connected paths. Note that the CPF algorithm relies on the optimal threshold criterion to effectively detect systematic defects, which can be determined by parameter-tuning or domain experts. The authors utilized domain experts to set a global threshold criterion of 12 for all defects, in which distances greater than 12 are recognized as systematic defects. The advantage of the CPF algorithm is that the threshold criterion allows for the detection of the *Scratch* defect. The limitation of applying a global threshold criterion for all defect-types is that the local spatial information, such as defective die density and distribution, defect-type geometry, and disjoint connection paths [due to random defects], is not considered. Specifically, the defective dies that are not associated with a connected path are completely disregarded. Additionally, defining a universal threshold for all defect-types cause scalability issues for real-world applications as the onset of complex, and mixed-type defects would require domain experts and frequent updates to threshold values.

To address the limitations of the CPF algorithm, the graph-theoretic approach for adjacency clustering (AC) was developed by Ezzat et al. (2021). Based on graph theory, this algorithm represents the dies and the neighborhood connections on the wafer map as the graph nodes, and graph edges. Although the AC algorithm is executed as a spatial clustering task, it functions as a spatial filtering method by leveraging spatial correlation information between adjacent dies to cluster the defective dies into two groups: random and systematic defects. The authors compared the AC and CPF algorithms, and demonstrated the improved performance of AC in filtering high complexity defects, and overall improved impact to the defect recognition task. The authors have noted that too small or too large separation loss would result in undesired filtering results (i.e., weak to absent filtering effect, same label wafer maps), and cross-validation may be used to determine the optimal weight trade-off. In comparison to existing preprocessing methods, this algorithm fully utilizes the available spatial information (i.e., spatial dependency of adjacent dies), demonstrating state-of-the-art performance.

Supervised learning

Supervised learning utilizes labels as a supervisory signal for training. Early literature for wafer map defect detection mostly consists of supervised machine learning algorithms, including common models such as artificial neural network (ANN), random forest (RF), and support vector machines (SVM). Note that in wafer map defect detection applications, multi-class classification is more popular and widely developed than multi-label learning. The methodologies discussed in this section are structured into three categories: (i) conventional machine learning, (ii) deep learning, and (iii) specialized modules.

Conventional machine learning algorithms used for WMDD include SVM, decision trees, and ensembles. Although a bit antiquated due to the onset of neural networks and deep learning, conventional ML algorithms can remain competitive. In related works, SVM and decision trees were prominently used for single-type WM defect classification as the classifiers are relatively computationally inexpensive, stable, and can work well with high-dimensional data (Chang et al., 2012; Hsu & Chien, 2007; Kim et al., 2020b; Li & Huang, 2009; Liao et al., 2014; Ooi et al., 2013). These methods reported a high overall detection accuracy (approximately > 90%), however demonstrated low detection rates for geometrically complex defect patterns (i.e., *Donut*, *Scratch*, mixed-types), and diminished effectiveness with imbalanced datasets. To boost overall classification accuracy, Jin et al. (2020) incorporated error-correcting output codes (ECOC) and SVM for single-type WM defect classification using CNN-based feature extraction.

Yu and Lu (2016) proposed the joint local and non-local linear discriminant analysis (JLNLDA) framework, which utilizes manifold learning to extract highly discriminative features. With the aim to preserve defect geometry at lower dimensional space, four neighborhood graphs: two graphs for local and non-local spatial information, and two penalization graphs that apply penalties to promote maximizing between-class separation, and minimizing within-class separation. Geometry, gray, texture, and radon-based features were generated, followed by dimensionality reduction and feature extraction. For wafer defect detection, JLNLDA was extended to construct JLNLDA-FD, a Fischer discriminant-based recognition model to compute the discriminant function value of a wafer map belonging to the defect classes, such that wafer maps are classified as the defect class with the maximum probability.

Saqlain et al. (2019) proposed a soft voting ensemble (SVE) classifier for wafer defect recognition and classification. Using the WM-811K dataset, three multi-type features (geometry-based, density-based, radon-based) are extracted, and used as inputs to train the base classifiers of the ensemble. The authors used four state-of-the-art ML classifiers for the ensemble: logistic regression, gradient boosting machine (GBM), ANN, and random forest. To train the proposed ensemble, the base classifiers are trained individually using the extracted features, and then in a soft voting ensemble approach, the results of the base classifiers are combined to output the final defect prediction. Soft voting uses weighted averages to determine the final prediction; based on performance, better performing classifiers have higher weights for voting. The authors reported defect classification accuracy of 95.87%, proving the ensemble classifier achieves improved performance relative to a single individual base classifier. Although both JLNLDA and SVE achieved high defect classification rates, their performance is contingent on manually generated features, which can bottleneck performance.

Extensions of supervised ANNs have featured in WMDD literature, including multilayer perceptron (MLP), and general regression network (GRN) (Adly et al., 2015a, 2015b; Huang, 2007; Huang et al., 2009; Tello et al., 2018). In (Huang, 2007) and (Huang et al., 2009), self-supervised MLP models were trained to recognize clusters of defective dies, however, classification was restrained to predicting good and bad wafers, such that limited details of the defect were learned. GRNs utilize Gaussian kernels as activation functions in the hidden layer. Adly et al., (2015b) applied a randomized bootstrapping technique to train an ensemble of GRN models, such that each model would learn from a random, independently sampled data to decrease variance, and increase detection accuracy. Similarly, Adly et al., (2015a) extended the previous work with a data dimensionality reduction technique, which employed Voronoi diagrams for data partitioning and K-means for clustering to represent the data at a reduced size. As the Voronoi diagrams portion the data into a vector space; smaller regions reflect different defect patterns, and K-means clustering was used to find the centroid for each region in the vector space, which was subsequently used for training. Both the GRN-based models demonstrated high accuracy, but by applying the data reduction technique, computational time complexity was reduced. As these previous works considered only single-type defects, Tello et al. (2018) combined the randomized GRN (RGRN) model with a CNN model. By using information gain theory to separate the data into single-type and mixed-type defects, RGRN and CNN classify single-type and mixed-type defects respectively, achieving an overall accuracy of 86.17%. Although mixed-type defect detection was investigated, a limited range of mixed-type defects were considered.

Deep learning models employ CNNs and additional layers for training. Due to the automated feature extraction capability, deep learning has been heavily applied for image-based tasks, including wafer map defect recognition and classification. Deep learning models typically have more than three layers, and with each progressive layer, the model extracts higher level features. Many related works utilize CNNs for single-type, and mixed-type WMDD. In (Batool et al., 2020; Bella et al., 2019; Du & Shi, 2020; Kim et al., 2020a; Maksim et al., 2019; Nakazawa & Kulkarni, 2018; Yu et al., 2019a), CNNs with customized model architecture were trained for single-type WM defect classification. For example, the custom CNN architecture by Nakazawa and Kulkarni (2018) for multi-class defect pattern classification achieved an overall test accuracy of 98.2%, and considered 22 defect classes (Fig. 6), in which many classes were variations of fundamental defect patterns. It should be noted that in the case of class distinctiveness, many classes were quite similar, such that misclassification rates were high as the model had difficulty differentiating between the similar-looking defect patterns. Additionally, in multi-class classification methods, mixed-type defect detection is difficult as the most salient defect pattern is typically predicted, disregarding the other present defects.

The related works for mixed-type WM defect detection framed the problem as multi-label classification (Devika & George, 2019; Hyun and Kim (2020); Wang et al., 2020) or multi-class classification (Byun & Baek, 2020; Kim et al., 2021; Kong & Ni, 2019, 2020b; Kyeong & Kim, 2018; Zhuang et al., 2020). For multi-label classification of mixed-type defects, CNN models used sigmoid activation to compute the probability for each defect label. On the other hand, for multi-class classification of mixed-type defects, Kyeong and Kim (2018) proposed the use of CNNs for mixed-type defect pattern classification by training multiple binary CNNs (Fig. 7). Each CNN is built to detect the absence or presence of a distinct pattern (*Scratch*, *Ring*, *Circle*, *Zone*), and then the CNN outputs are combined. By leveraging multiple CNNs, this method has the advantage of adaption, as new defect patterns can be easily trained and added to the existing framework. Compared to SVM and multilayer perceptron (MLP), the proposed CNN achieved superior classification accuracy, recall and precision of 0.910, 0.945, and 0.949 respectively. Similarly, in (Zhuang et al., 2020), a network of deep belief networks (DBN) was used to classify six defect patterns for single-type and mixed-type defect classification. Kong and Ni proposed mixed-type defect detection by pattern segmentation, such that overlapped defect patterns are processed into multiple single patterns, which are then classified using multiple binary CNNs (Kong & Ni, 2019, 2020b). Both proposed models achieved comparable classification performance as other high performing models and demonstrated how pattern segmentation of overlapped mixed-type defects can improve recognition and classification accuracy. Kim et al. (2021) applied the object detection algorithm, single shot detector (SSD), to effectively recognize, segment and classify the multiple instances of defect patterns within a mixed-type defect sample. As object detection frameworks require bounding box (BB) information (for the desired object instances), an automatic BB generator was designed to utilize digital image preprocessing techniques and libraries (i.e. *PIL,* spatial filters) to obtain the BBs. The SSD algorithm simultaneously solves the object classification and localization problems, which subsequently improves run-time, and performance. The SSD model utilized pretraining from large-scale image datasets, and fine-tuned the last output layer on a selection of the WM-811K data. Compared to the CNN model, the proposed SSD model achieves a higher accuracy for single-type and mixed-type defects.

The methods categorized as specialized modules integrate advanced model elements different from standardized model components, which can encompass specialized loss functions, modified kernel functions, etc. Park et al. (2020) proposed a Siamese network integrated with an uncertainty-reducing technique for class label reconstruction via G-means clustering (Fig. 8). For discriminative feature learning, the Siamese network learns feature embeddings based on similarities between the input image pairs, and aims to minimize the contrastive loss, such that embeddings for similar images are closer together, and embeddings for dissimilar images are farther apart. G-means clustering leverages the

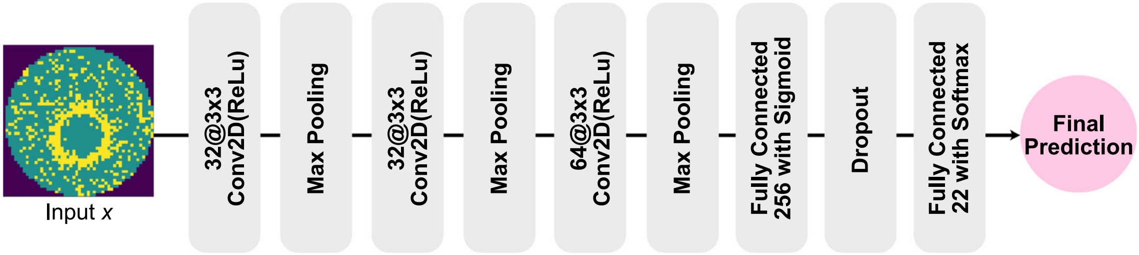
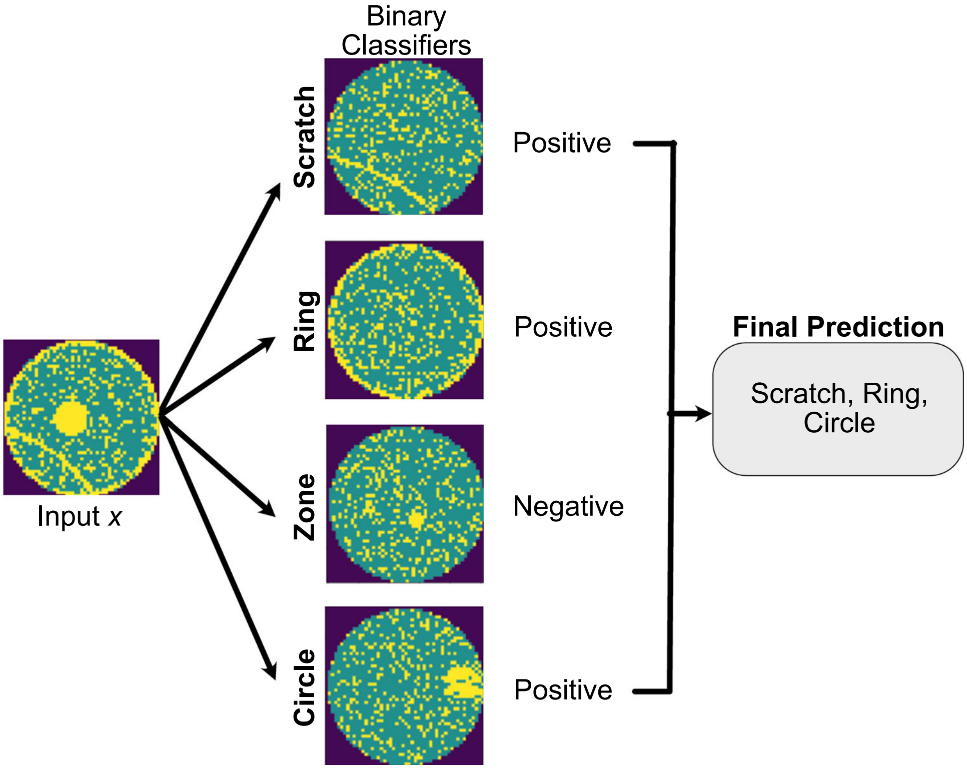


Fig. 6 Structure of proposed CNN by Nakazawa and Kulkarni (2018)

learned feature embeddings from the Siamese network to enable enhanced class label reconstruction and outlier detection. The results demonstrate that the proposed model can segment mixed-type defects, however, has difficulty with controlling the degree of pattern segmentation, and differentiating between the unknown cases from the known cases. By leveraging class label reconstruction, uncertainty associated with the wafer map labels can be mitigated.

Modified convolutional blocks were proposed by Wang et al. (2020), Tsai and Lee (2020a), Hyun and Kim (2020), and Alawieh et al. (2020). Wang et al. (2020) used deformable convolution networks (DCN) for multi-label classification, which demonstrated enhanced performance as deformable convolutional layers can learn and recognize the geometric variations of defect patterns. Deformable convolutional units learn the two-dimensional offsets to learn different deformations of the filter sizes and geometric characteristics, which are subsequently added to a standard convolution (Fig. 9) (Dai et al., 2017; Zhu et al., 2019). The authors compared the proposed DCN to state-of-the-art mixed-type defect classification models on the Mixed WM-38 dataset, in which the results demonstrated the superior performance of DCN in the detection of complex mixed-type defects. Similarly, Tsai and Lee (2020a) incorporated depth-wise separable convolutions to improve run-time and reduce overfitting as they have fewer parameters than standard convolutions. By using depth-wise separable convolutions, the proposed model achieved a 96.63% classification accuracy based on single-type defect patterns. Another development of modified convolutional blocks was introduced by Hyun and Kim (2020); a memory module to keep track of a fixed number of rare occurrences for each class to mitigate class imbalance issues. The memory module is used to learn high quality representative samples in latent space for each defect class.

Fig. 7 Proposed mixed-type defect classification model in Kyeong and Kim (2018)



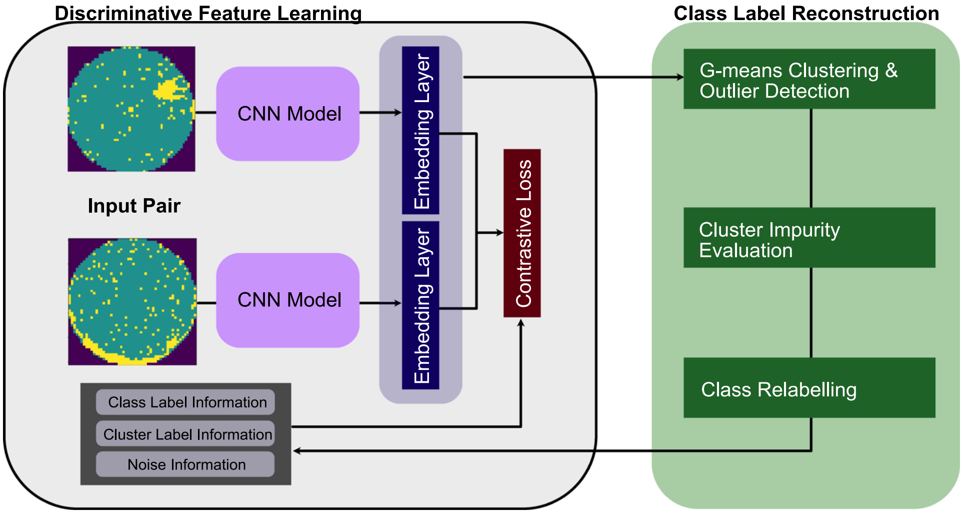

Discriminative Feature Learning
Class Label Reconstruction
CNN Model
Input Pair
CNN Model
Embedding Layer
Embedding Layer
Contrastive Loss
Class Label Information
Cluster Label Information
Noise Information
G-means Clustering & Outlier Detection
Cluster Impurity Evaluation
Class Relabelling

Fig. 8 Proposed Siamese network with class label reconstruction in (Park et al., 2020)

To learn the low dimensional representations of the data within the CNN structure, this method utilized triplet loss for training. Compared to CNN and SVM variations, the proposed memory module achieved comparable test accuracies on three different datasets.

Alawieh et al. (2020) proposed a reject option for CNN deep selective learning, such that misclassification of unknown defects can be avoided. Deep selective learning is leveraged when new defects emerge, change in class distribution, and resource allocation. The model is trained to achieve an optimal trade-off between rejection and classification, such that the model rejects prediction of select wafer maps when the risk of misclassifying is high. This creates a pool of samples to be examined for enhance understanding and identification of new defects. The authors demonstrated the use of deep selective learning can achieve superior performance relative to conventional CNNs. Similarly, Cheon et al. (2019) designed a CNN model with an unknown defect detection option. Using kNN, the anomalies (unknown defects) are recognized, compared to other known defect clusters, and classified as unknown when determined as lacking cluster membership.

Unsupervised learning

To leverage the abundance of unlabeled data, unsupervised learning has been applied for clustering, as well as the auxiliary task of pre-training to supplement supervised learning. In context of wafer map defect detection, the related unsupervised learning works are introduced.

Clustering algorithms utilize similarity or distance measures to group data; they aim to minimize the distance between intra-cluster samples (high intra-cluster similarity), and maximize inter-cluster distances (low intra-cluster similarity). Spatial clustering is applied for wafer map segmentation, such that the defect patterns are separated into clusters. In early works that utilized clustering algorithms, adaptive resonance theory (ART) based models (Chen & Liu, 2000; Choi et al, 2012; Hsu & Chien, 2007; Palma et al., 2005) were prominently used. These ART-based models are recurrent models, demonstrating memory retention, knowledge adaption and growth when identifying characteristics of new or similar defect patterns. In (Taha et al., 2018), spatial dependence across all maps was considered for the proposed wafer clustering algorithm, Dominant Defective Patterns Finder (DDPfinder). Similarly to Adly et al., (2015a), Voronoi diagrams are used to partition the defect

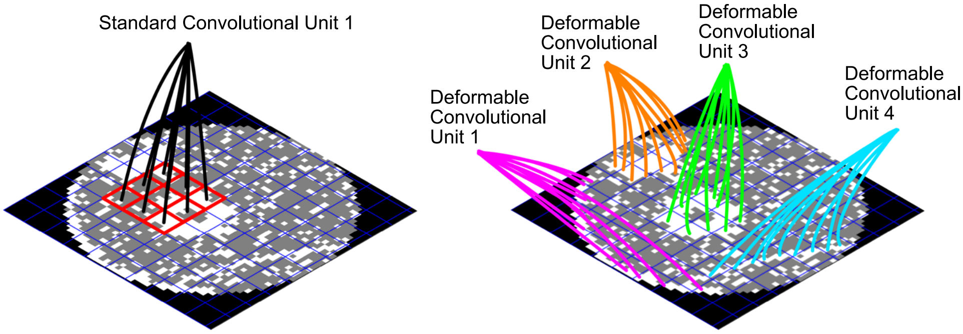

Standard Convolutional Unit 1
Deformable Convolutional Unit 1
Deformable Convolutional Unit 2
Deformable Convolutional Unit 3
Deformable Convolutional Unit 4

Fig. 9 Standard convolution sampling (left) and deformable convolution sampling (right)

patterns, and to determine the respective spatial dependence relative to the identified centroid defective die point. Hierarchical clustering was used by Alawieh et al. (2018) to minimize clustering sensitivity to outliers; incorporating various optimization methods to determine the optimal number of clusters, optimal number of singular values for noise removal, and optimal number of defect patterns. As clustering algorithms are sensitive to initialization and hyperparameters (i.e., number of clusters), many suffered from difficulty of determining the appropriate number of clusters for defect patterns (Patel et al., 2015; Xu & Tian, 2015).

Related works (Hwang & Kim, 2020; Jin et al., 2019; Kim et al., 2018) leveraged clustering algorithms for defect detection. Kim et al. (2018) utilized connected-path filtering, and then spatial clustering via infinite warped mixture models (iWMM). iWMMs (originally introduced in (Iwata et al., 2013)) apply a warping function to the defect clusters, such that in the latent space, the clusters have Gaussian shapes. In (Ezzat et al., 2021; Iwata et al., 2013; Kim et al., 2018), the authors report iWMM as an effective clustering algorithm due to its warping function, and ability to effectively estimate the number of clusters, which circumvents the influence of setting the number of clusters. However, Kim et al. (2018) noted that iWMM had difficulty in appropriately isolating the partial-ring defect pattern due to its complex and non-Gaussian geometry. Jin et al. (2019) introduce DBSCANWBM, a novel DBSCAN-based clustering method. DBSCANWBM inherits DBSCAN characteristics, and was adapted to: (i) consider defect-type for outlier detection, (ii) bypass the requirement to specify number of clusters, (iii) parallelize outlier detection and defect detection, and (iv) detect both single-type and mixed-type defects. By adjusting outlier removal relative to defect-type, the systematic defect geometries can be better preserved, which in turn, can improve classification accuracy. Hwang and Kim (2020) developed a one-step clustering method that combines Gaussian mixture models and Dirichlet process (DP) to a VAE framework. Within the proposed VAE framework, DP is used to automate the updating of number of clusters, and the GMMs are employed as a prior distribution to learn the nuances of different wafer maps. Like iWMM, and DBSCANWBM, this VAE framework works without specifying the number of clusters in advance. The VAE framework encodes and decodes latent feature representations that follow a Gaussian mixture distribution (Hwang & Kim, 2020). The authors reported that their proposed clustering framework estimated the number of clusters more accurately than the comparison models, and achieved better clustering performance relative to adjusted mutual information and adjusted rand index. The clustering methods that utilized generative models have demonstrated improved performance as the models are built to learn effective feature representations.

Unsupervised pre-training is typically conducted by training an autoencoder in an unsupervised approach to minimize the reconstruction loss and learn latent feature representations of the data. For classification tasks, a classifier is added to the trained encoder and fine-tuned; the fine-tuning adjusts the encoder and classifier. The general process of unsupervised pre-training is shown in Fig. 10. Shon et al. (2021) applied unsupervised pre-training and data augmentation to improve CNN classifier performance based on limited labeled wafer maps. Using the unlabeled data of WM-811K, a convolutional variational autoencoder (CVAE) was trained in efforts to better initialize the feature extraction layers of the CNN classifier. Subsequently, the CVAE encoder and CNN classifier are fine-tuned in an end-to-end manner by minimizing the cross-entropy loss. The results showed that the proposed method achieved high classification performance at early epochs, indicating the benefit of unsupervised pre-training. Although pre-training can improve downstream classification performance, as the WM-811K data consists of single-type defects, the proposed model is limited in complex mixed-type defect recognition as CVAE may have difficulty differentiating between the multiple defects with a single discriminative network. Similarly, Yu (2019) proposed a two-phase methodology for wafer map recognition: an enhanced stacked denoising autoencoder (ESDAE) for feature learning via unsupervised pre-training, and then supervised finetuning. ESDAE consists of two autoencoders, which incorporate manifold regularization such that intrinsic local and nonlocal geometric information is preserved. ESDAE involves a cost-sensitive layer-wise training procedure, in which each layer is trained to minimize the reconstruction error, and assigns different costs to different defect classes for misclassification to address class imbalance. The experimental results on the influence of manifold regularization demonstrate that performance improved with the increasing degree of regularization (γ). Compared to a typical stacked denoising autoencoder (SDAE), logistic regression, DBN, and back propagation network (BPN), ESDAE achieved the best defect recognition accuracy of 97.03%. Despite the improved performance, the proposed methodology involves feature generation of original geometrical, gray, texture, and projection features for model training; generally, as it is difficult to estimate the effectiveness of manually generated features, model performance may be hampered. Likewise to CVAE, ESDAE trains on the single-type defects in WM-811K, and as such is inadequate against mixed-type defects.

Semi-supervised learning

The performance of supervised learning is limited by the amount of available labels; on the other hand, without the supervisory signal from labels, the performance of unsupervised learning for defect classification is unsatisfactory

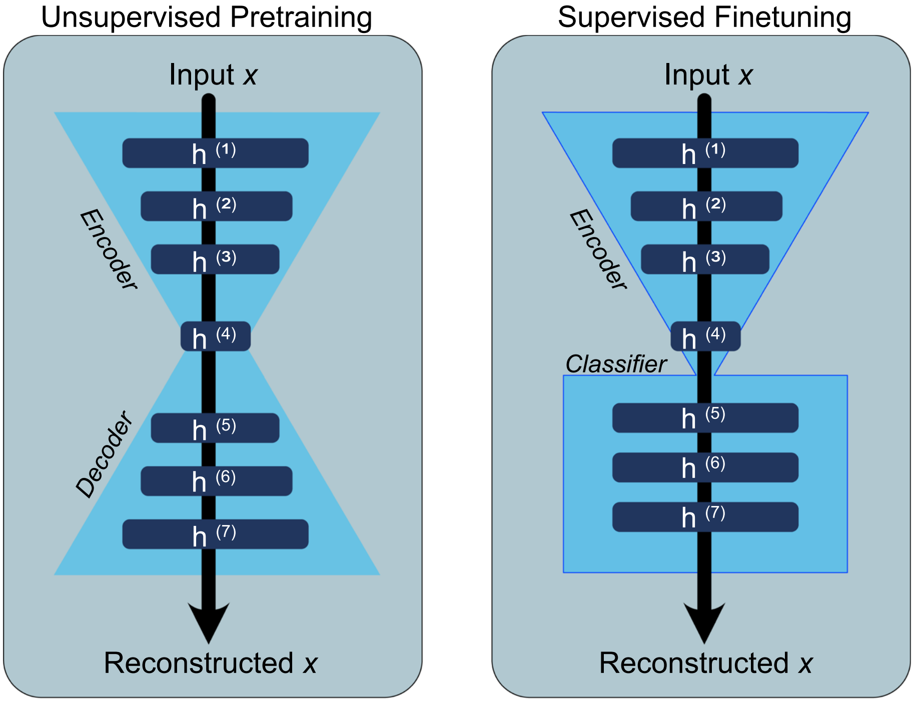


Fig. 10 General unsupervised pretraining using layer-wise training of autoencoder with layer-wise training (left) and supervised finetuning with classifier (right)

in comparison. Semi-supervised learning is introduced, and addresses the limitations of supervised and unsupervised learning. Regarding real-world applicability, with limited available labels, a surplus of unlabeled wafer maps, and large volumes of incoming unlabeled wafer maps, semi-supervised learning can achieve better performance as it utilizes both labeled and unlabeled data for model training. For semi-supervised learning, the labeled wafer maps are used to learn the relevant features for each defect pattern, and then the unlabeled wafer maps are used to refine the feature representations. To the best of our knowledge, semi-supervised learning algorithms for wafer map defect recognition and classification have been scarcely developed.

Kong and Ni (2018) trained a CNN-based Ladder network in a semi-supervised manner to detect and classify wafer map defects. The semi-supervised Ladder network consists of a clean encoder, a corrupted encoder, and a decoder, which were trained and tested separately on two datasets with 22 classes of single-type defect patterns. The encoders are responsible for learning the latent features of the wafer maps. The latent features from the encoder layers are shared with the decoder through skip connections to recover additional spatial information. Given the noised latent features from the corrupted encoder, the decoder reconstructs the wafer maps with the aim to minimize the reconstruction error at each layer. Compared to supervised CNN with varying amounts of labeled data, the authors established how semi-supervised learning can improve wafer map defect classification accuracy. As the proposed framework trained on two small datasets containing only single-type defect patterns, the small class sample sizes most likely skewed feature learning, such that the model had difficulty differentiating between similar-looking pattern variations. This is shown by the confusion matrices reported in (Kong & Ni, 2018), which divulge the misclassification rates of select defects. Additionally, as the datasets contained only single-type defect wafer maps, defect classification is limited and requires modification and model re-training for mixed-type defects.

In (Yu & Liu, 2020), PCACAE, a novel semi-supervised two-dimensional PCA-based convolutional autoencoder with effective feature extraction capability is introduced. To overcome class imbalance and preserve spatial information, conditional two-dimensional PCA (C2DPCA) is proposed. C2DPCA aims to find the optimal projection direction by minimizing the reconstruction error, and as an image projection method, can effectively map the high dimensional wafer maps into lower dimensional space. By transforming the principal eigenvectors from 1 to 2D, C2DPCA-based kernels are formed, such that discriminative principal components are learned and used downstream for pretraining and finetuning purposes. The authors compared PCACAE performance to a pretrained deep learning models (i.e., AlexNet, GoogleNet), stacked denoising autoencoder (SDAE), and DBN. The results and visualizations reported in (Yu & Liu, 2020) indicate the usefulness of pretraining, and that the C2DPCA-based kernels have effective, and powerful feature learning capabilities. As the PCACAE framework trained on the WM-811K dataset, defect recognition is limited to single-defect patterns. With the use of pretraining, PCACAE

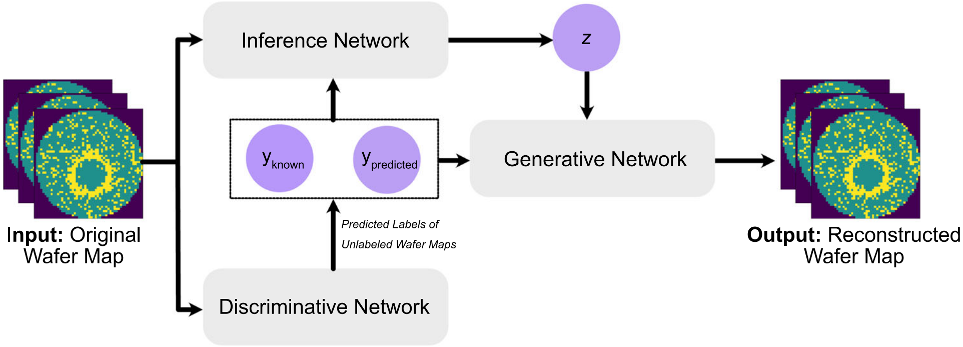

Inference Network
z
Generative Network
Discriminative Network
y_{known}
$y_{predicted}$
Predicted Labels of Unlabeled Wafer Maps
Input: Original Wafer Map
Output: Reconstructed Wafer Map

Fig. 11 Proposed SVAE methodology in (Kong & Ni, 2020a)

has shown reduced computational run-time (per iteration) relative to the comparison models. Although C2DPCA has demonstrated to be effective, it is limited regarding non-linear data, as it is essentially an orthogonal linear transformation on the data.

Kong and Ni (2020a) also presented a semi-supervised variational autoencoder (SVAE) with incremental learning (Section Enhanced learning strategies) for wafer map defect classification, which was trained and tested on two datasets with 22 classes of single-type defect patterns. The proposed SVAE framework (Fig. 11) comprised of three networks: (i) inference network, (ii) discriminative network, and (iii) generative network. The inference network is responsible for approximating and learning the latent feature representations of the wafer map defects. The discriminative network is used to predict the labels of the unlabeled WMs, including WMs with rare/unseen defect patterns. The generative network leverages the learned latent features and predicted labels for the unlabeled wafer maps to reconstruct the original wafer map. The authors compared the classification performance of a CNN, and the supervised components of SVAE, and semi-supervised Ladder network (Kong & Ni, 2018) with different percentages of supervised training data. The results demonstrated the superior performance of the semi-supervised approach as the Ladder network, and SVAE consistently achieved higher classification accuracy than the supervised CNN, particularly at lower percentages of supervised data. Despite the improved performance, the confusion matrices showed some defect classes were prone to misclassification, which may have been attributed by the class imbalance as the classes of the datasets were the defect patterns and their respective variants. Yu et al. (2019b) proposed a hybrid learning model, stacked convolutional sparse denoising autoencoder (SCSDAE). Employing data sampling methods, SCSDAE has demonstrated effective learning of discriminative features from the single-type WM data; with performance superior to deep neural networks. Similarly to (Kong & Ni, 2018), the training and test datasets contained only single-type defect patterns, which constrains defect recognition and classification to single-type defects, disregarding the onset of mixed-type defects.

A semi-supervised convolutional deep generative model (SS-CDGMM), shown in Fig. 12, was proposed by Lee and Kim (2020). In contrast to other semi-supervised models which established multi-class classification for single-type defect patterns, a multi-label configuration for mixed-type defect classification was utilized. Kingma et al. (2014) introduced new semi-supervised deep generative models (SS-DGM), wherein the data is described as being generated by a latent class variable and a continuous latent variable. As an extension of SS-DGM, SS-CDGMM consists of multiple discriminative networks structures, such that each corresponding latent class variable is dedicated to one of the fundamental defect-types. Like Kong and Ni (2020a), SS-CDGMM consists of an inference network, discriminative networks, and a generative network, however each discriminative network is used to learn the absence and presence of its respective single-defect pattern. Compared to various models (i.e., CNN, multi-layer perceptron (MLP), SS-DGM, unified VAEs), including the state-of-the-art, convolutional ladder network (ConvLadder), the results showed comparable or better performance to the state-of-the-art. Relative to the comparison models, SS-CDGMM demonstrated how it effectively uses labeled and unlabeled data, as well as the effectiveness of using multiple discriminative networks. However, as the training and test data were generated and balanced across the classes, the impact of class imbalance has not been investigated or addressed. Additionally, only four distinct single-type defect patterns were considered, disregarding the other known distinct defect patterns (i.e., Donut, Near-full). Although more defect patterns may be considered, this would result in higher run-times as the marginal log-likelihood component of the objective function requires computation over all defect classes.

Moving away from generative modelling, self-supervised pretraining is emerging as an effective pretraining method for semi-supervised frameworks and classification tasks (He et al., 2019; Chen et al., 2020b). Self-supervised contrastive

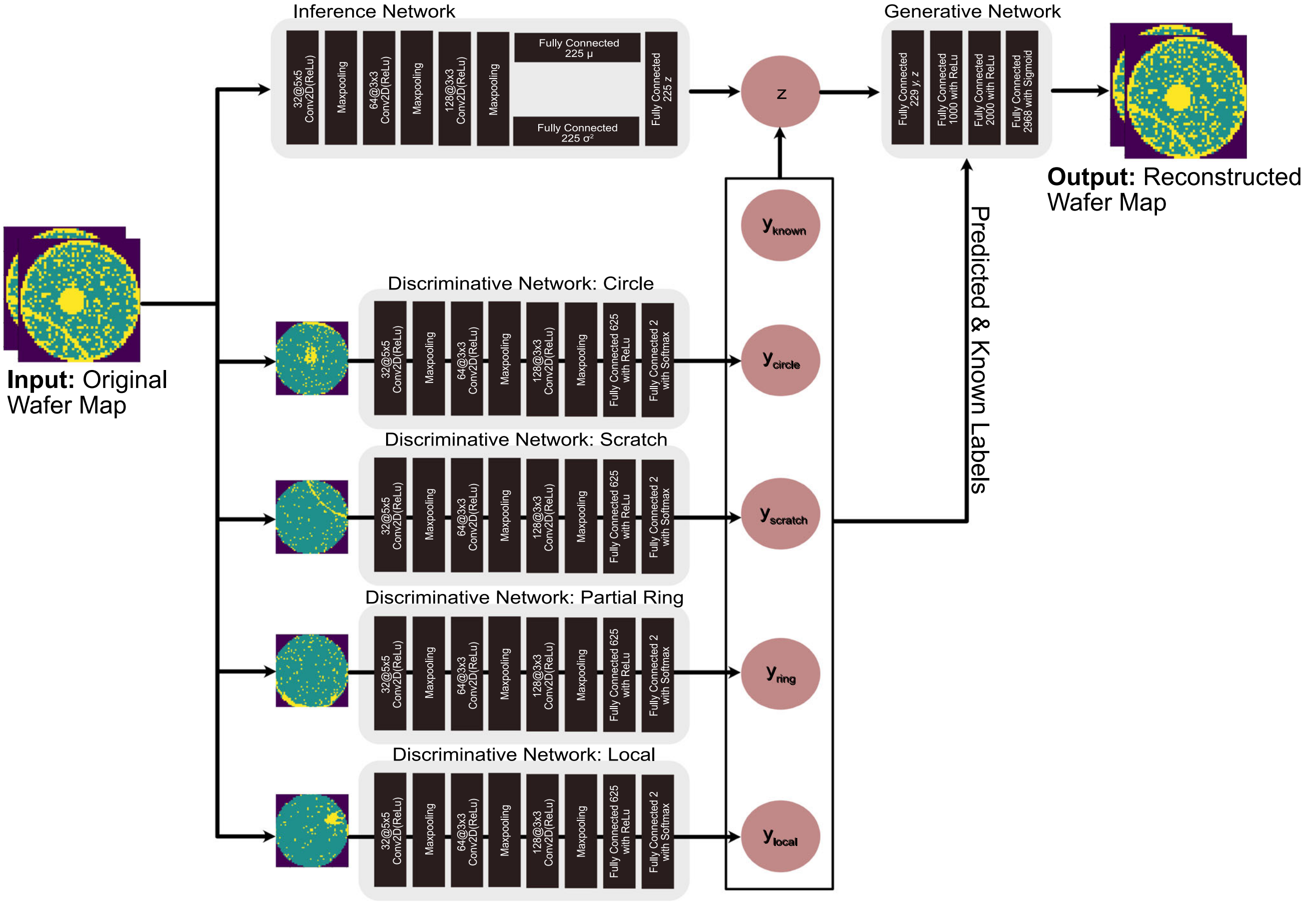


Fig. 12 Proposed SS-CDGMM in (Lee & Kim, 2020)

learning has been increasingly leveraged as a feature learning method, wherein meaningful representations can be learned from unlabeled data and data augmentations. Hu et al. (2021) proposed a contrastive learning framework for single-type defect patterns, followed by supervised finetuning of a classifier. Despite performance that is lower than other algorithms, the reported results demonstrate detection rates on par with state-of-the-art contrastive methods (i.e., SimCLR), and great potential for contrastive learning.

Enhanced learning strategies

The methods included in this section focus on enhancing model learning, and are used to elevate model performance. They are organized into the following groups: (i) data augmentation, (ii) incremental learning, (iii) transfer learning and fine-tuning, and (iv) model optimization.

Data augmentation aims to reduce overfitting by increasing the amount of data, and is typically used to mitigate class imbalance issues, which neural networks and deep learning models are particularly sensitive towards (Perez & Wang, 2017). Data augmentation can be executed in many ways, such as resampling, data modification, and data generation.

Resampling methods function to balance the class distribution of the existing data. Undersampling and oversampling are subcategories of resampling methods. Undersampling reduces the amount of data examples from the majority classes by removing data, whereas oversampling increases the amount of data examples by sampling from the minority classes with replacement. Both subcategories of resampling methods are effective for obtaining a more balanced class distribution, however, have their share of limitations. Undersampling ultimately reduces the overall amount of data, and may disregard critical data examples for the majority classes, which may impede feature learning and model performance. Oversampling may result in overfitting and increased generalization error, as well as increased computational time as the overall amount of training data is increased. Due to the limitations of resampling methods, data augmentation via modification and generation are typically conducted as they can increase and balance the amount of data, as well as increase the data diversity.

Data modification methods apply label-preserving operations to create synthetic variations of the existing data. Considering the circular shape of the wafer maps and the diversity of defect patterns, select geometric operations can be applied to maintain the geometric characteristics, and original labels. In Kang (2020) and Jang et al. (2020), rotation and horizontal flipping operations were applied to create diversified, rotation-invariant wafer maps, which subsequently improved defect classification performance. Similarly, Saqlain et al. (2020) applied random rotations of 10°, horizontal flipping, width shift, height shift, shearing, channel shifting, and zooming to augment the data. These operations are used as they diversify the data with changes in orientation, position, and/or size. The different operations used in data modification methods help improve model generalization as models are trained to be highly tolerant to the diversified variations of defect patterns.

The data generation methods utilize generative models, such as generative adversarial networks and autoencoders, to supplement the existing collection of data by generating new synthetic data. The generative models focus on learning the latent feature representations and distributions of the data. As the performance of many deep learning models are contingent on the amount and distribution of labeled data, data generation methods are used to create realistic, new instances of data. GANs consist of two convolutional neural networks: a generator and discriminator (Fig. 13). The generator learns to create authentic fake data, and the discriminator learns to distinguish between the real and fake data. Variations of GANs have been developed to improve the generative modelling capability. Wang et al. (2019) proposed the adaptive balancing generative adversarial network (AdaBalGAN), a conditional categorical GAN that incorporates imbalanced learning to generate a balanced set of synthetic data. In addition to the generator and discriminator, AdaBalGAN includes an adaptive generative controller, which recognizes the minority defect classes by considering defect class size, as well as the recognition accuracy difference between each defect class and the majority defect class. By recognizing the imbalanced class distribution, the adaptive generative controller automatically adjusts the number of synthesized wafer maps for each defect-type. Ji and Lee (2020) developed a deep convolutional GAN, which compounds the image processing capabilities of multiple convolutional layers, for data augmentation. Aside from GANs, there are many types of autoencoders, including variational, convolutional, denoising, stacked, and sparse; the fundamental components of autoencoders are the encoder and decoder. The encoder compresses the input into latent space representation, and the decoder uses the latent representation to reconstruct the input. Shawon et al. (2019) and Tsai and Lee (2020b) utilized a convolutional autoencoder (CAE) to generate new instances of denoised training data to improve model training of deep convolutional neural networks. Similarly, in (Lee & Kim, 2020), the authors employed the trained VAE to generate labeled wafer maps by leveraging the learned class latent variables for each defect-type. Data augmentation via generation can create highly diverse and realistic data, however, requires substantial computational time and power to effectively train the generative models.

Incremental learning (IL) aims to increase model performance by extending and adapting an existing model's knowledge base with new training data. In context of real-world wafer fabrication, labels are expensive to obtain, and have limited availability, which bottlenecks model performance. Additionally, as defect complexity evolves and new, unseen defect patterns emerge, model efficiency may decrease overtime. As such, IL methods are employed to enhance model performance in the long-term against evolving wafer map data and defect patterns. Popular methods include active learning, and pseudo-labeling.

Active learning utilizes a querying strategy to select informative unlabeled data for manual annotation to fine-tune and further train an existing model (Fig. 14). It is important to note that there are many querying strategies (Settles, 2009), including uncertainty sampling, information gain, query-by-committee, expected error reduction, and total expected variance minimization. Shim et al. (2020) proposed a CNN with active learning via uncertainty sampling for wafer map defect classification. Uncertainty sampling selects the most ambiguous unlabeled data examples; least confidence, margin, and entropy are common estimators for uncertainty. In addition to the common uncertainty estimators, the authors compared mean standard deviation, variation ratio, Bayesian active learning by disagreement (BALD), and predictive entropy as uncertainty estimation methods. Their results indicated that BALD and mean standard deviation provided the best performance for defect classification via CNN with active learning. On the other hand, Kong and Ni (2020a) employed active learning using information entropy for their semi-supervised models, such that the unlabeled wafer maps with the maximum information entropy were selected for labeling and model fine-tuning. When investigating the significance of active learning, and pseudo-labeling, the results demonstrated improved classification accuracy. Although active learning strategies have helped in improving model performance, they are vulnerable to class imbalance, and catastrophic forgetting. Class imbalance introduces sampling bias in query sampling, which skews the querying towards the newer classes (Ren et al., 2020), and brings on catastrophic forgetting. In the process of finetuning the model with the new labeled data, catastrophic forgetting can occur when the previously learned information is degraded, and significantly lowers model generalization (Luo et al., 2020). The effectiveness of active learning methods is sensitive to

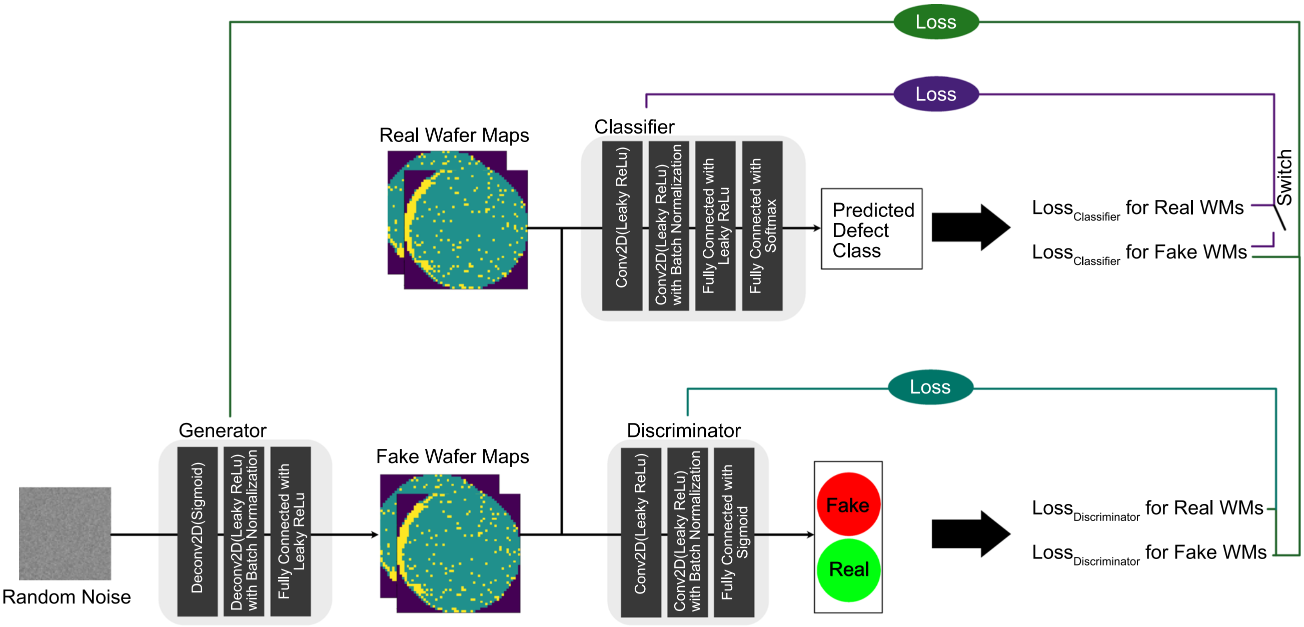
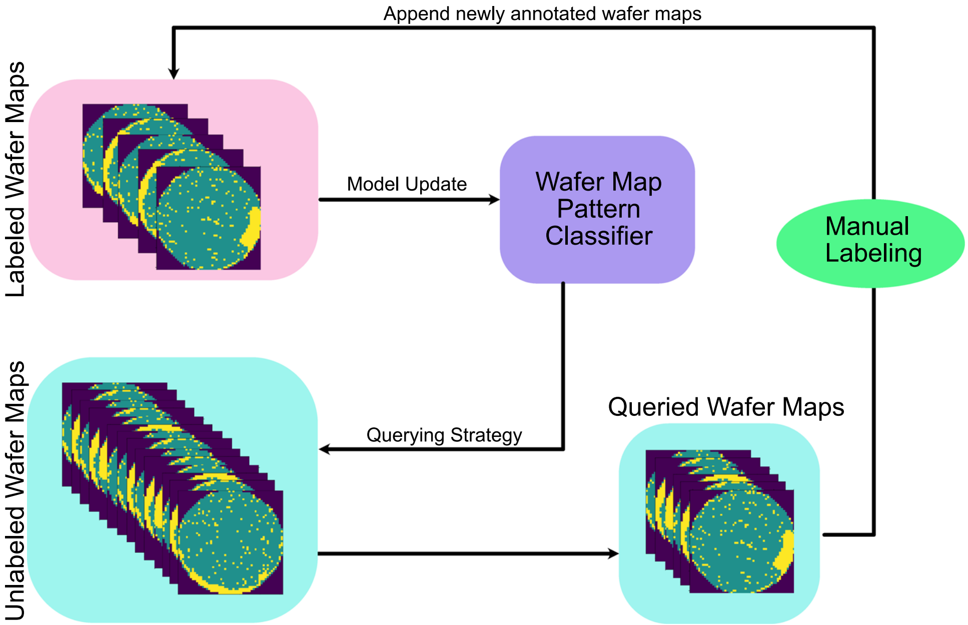

Loss
Loss
Real Wafer Maps
Classifier
Conv2D(Leaky ReLu)
Conv2D(Leaky ReLu) with Batch Normalization
Fully Connected with Leaky ReLu
Fully Connected with Softmax
Predicted Defect Class
Loss_Classifier for Real WMs
Loss_Classifier for Fake WMs
Switch
Loss
Generator
Deconv2D(Sigmoid)
Deconv2D(Leaky ReLu) with Batch Normalization
Fully Connected with Leaky ReLu
Fake Wafer Maps
Discriminator
Conv2D(Leaky ReLu)
Conv2D(Leaky ReLu) with Batch Normalization
Fully Connected with Sigmoid
Fake
Real
Loss_Discriminator for Real WMs
Loss_Discriminator for Fake WMs
Random Noise

Fig. 13 AdaBalGAN framework structure from (Wang et al., 2019)

Fig. 14 General procedure for active learning



the querying and model updating strategies, which warrants careful consideration for model implementation.

Pseudo-labeling supplements the incremental training data for model fine-tuning with predicted class labels for the unlabeled data. As a semi-supervised learning strategy, this method uses an existing trained model to assign the pseudo-label as the class with the maximum predicted probability. The pseudo-labels for the unlabeled data would increase the overall training dataset size, however, it is important to note that pseudo-labels may disturb model performance if they are incorrectly predicted. Kong and Ni (2020a) implemented pseudo-labeling with confidence level constraints. The authors computed and compared the information entropy for each unlabeled wafer map against a criterion threshold to ensure highly confident wafer maps were used for model fine-tuning. Similarly, to account for uncertainty, a 2:1 ratio for the original labeled wafer maps and pseudo-labeled wafer maps to diminish the potential disturbance from incorrect pseudo-labels.

Transfer learning is the process of utilizing a pretrained model for another task. As the pretrained models were trained

Table 5 Comparison of pretrained models for wafer map defect detection

	T-DenseNet (Shen & Yu, 2019)	VGG (Ishida et al., 2019)	Faster R-CNN-KITTI (Chien et al., 2020)	Faster R-CNN-COCO (Chien et al., 2020)
Center	64.52%	88.80%	98.92%	97.81%
Donut	91.18%	89.00%	–	–
Edge-Loc	81.46%	86.90%	–	–
Edge-Ring	66.79%	83.90%	–	–
Loc	100.00%	72.60%	97.27%	98.50%
Scratch	72.60%	77.20%	98.32%	96.14%
Near-full	99.31%	90.50%	–	–
Random	65.52%	73.50%	99.26%	99.06%
None	85.45%	95.60%	–	–

on large, diverse image datasets (i.e., ImageNet, CIFAR-10), it is presumed that the model effectively learned feature representations and obtained powerful generalization capabilities. The learned feature representations of the pretrained models can be re-purposed to train a new classifier, or the pretrained models can be fine-tuned to fit to a specific dataset and task. The application of transfer learning and fine-tuning can significantly reduce training time, and achieve high performance without requiring large volumes of data. Related works have utilized pretrained models for wafer map defect recognition and classification. Shen and Yu (2019) proposed the T-DenseNet framework; the pretrained DenseNet model was fine-tuned on the wafer map dataset, and then the refined feature representations were used to set up an online testing system for incoming unlabeled wafer maps. Similarly, the pretrained VGG model (Ishida et al., 2019), and faster R-CNN model (Chien et al., 2020) were utilized for wafer map defect recognition and classification. In Table 5, the performance of the models in (Chien et al., 2020; Ishida et al., 2019; Shen & Yu, 2019) for the test dataset is shown, and reflect how effective deep transfer learning is despite the shorter training times, and how pretrained model selection may affect performance on the downstream tasks.

Research has demonstrated the importance of model selection and hyperparameter tuning as these design choices (i.e., conditional variables, objective function, architecture, etc.) significantly influence performance (Banchhor & Srinivasu, 2021; Parsa et al., 2020; Ungredda & Branke, 2021). As an enhanced learning strategy, model optimization concentrates on the advanced strategies for optimizing model parameters. For image tasks like wafer map defect recognition and classification, the design choices for model architecture can affect performance and computational time. Standard optimization techniques involve extensive hyperparameter tuning of layer parameters (i.e., stride, filter size, etc.), training batches and epochs, etc., which typically requires extensive manual searching. As such, strategies for optimization policies and network architecture engineering have been developed to automate the design process.

Recently, reinforcement learning (RL) models have been leveraged to parse optimization policies. Bello et al. trains a recurrent neural network (RNN) controller with RL for neural optimizer searching (Bello et al., 2017). Essentially, the performance of child networks trained with different sets of optimizer update rules are compared to determine the optimal set of updating rules for optimization methods (Bello et al., 2017). Similarly, in (Shon et al., 2021), RL was used to train a RNN controller to determine the optimal data augmentation policy for wafer map transformation operations (i.e., rotation, flipping, zooming). The general training process for RNN controllers and search algorithms is shown in Fig. 15. Architecture engineering is used to learn and automate the design process of deep neural network design selection. Related works, like (Baker et al., 2017; Zoph et al., 2018) have also used RL to explore and discover high performing network architectures relative to task and dataset. In both applications, RL was leveraged for as a search algorithm for optimal parameters and design, which improved model training and performance, but required separate and extensive training.

In contrast to various existing frameworks for global optimization of hyperparameters and model parameters (i.e., grid search, random search, sequential search), Bayesian optimization (BO) frameworks have demonstrated state-of-the-art performance with high efficiency in computationally expensive-to-evaluate applications (Snoek et al., 2012, 2015). As a black-box method, BO algorithms aim to probabilistically model the unknown function—commonly with Gaussian processes—and establish the posterior distribution of the respective results for the explored hyperparameter settings. By maintaining the resulting posterior distribution and

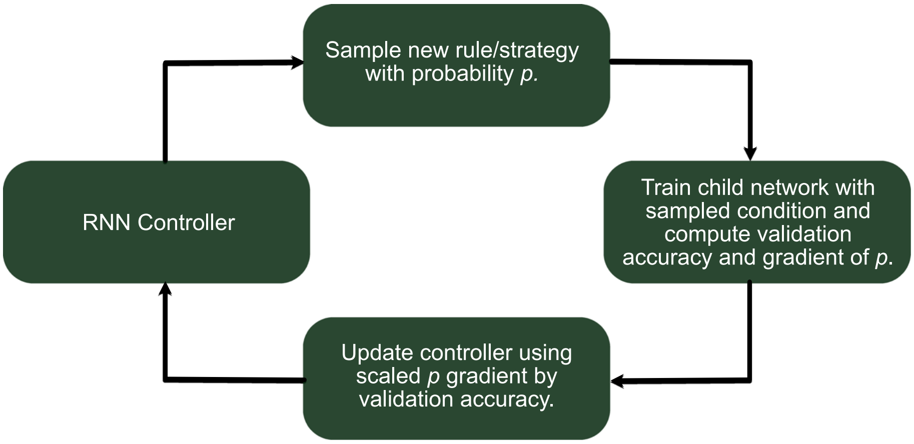

Sample new rule/strategy with probability *p*.
Train child network with sampled condition and compute validation accuracy and gradient of *p*.
Update controller using scaled *p* gradient by validation accuracy.
RNN Controller

Fig. 15 General training procedure for child networks and search algorithms

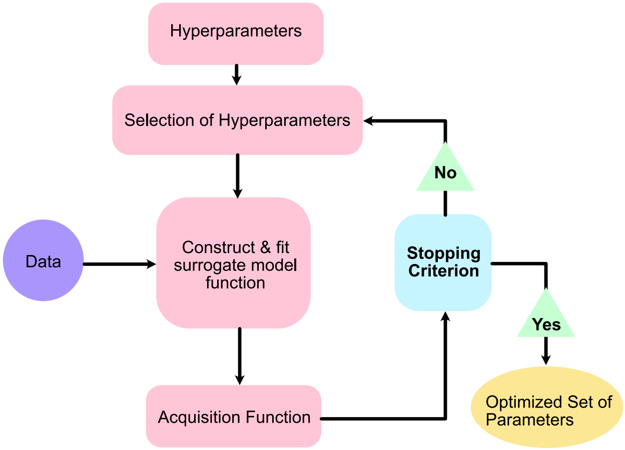

Hyperparameters
Selection of Hyperparameters
No
Data
Construct & fit surrogate model function
Stopping Criterion
Yes
Acquisition Function
Optimized Set of Parameters

Fig. 16 General flowchart of Bayesian optimization framework

exploiting past observations, BO algorithms utilize an acquisition strategy to make informed decisions about which best set of function parameters to evaluate next. As demonstrated in (Jang et al., 2020), Gaussian process-based BO was used to tune the CNN hyperparameters, such as learning rate, filter size, number of filters, and number of nodes in the fully connected layers. The hyperparameter settings were evaluated on the training data, and considered training time to prevent overfitting attributed by high complexity model architecture. In Fig. 16, the flowchart of the general BO framework is shown.

Discussion and conclusion

The current status of semiconductor wafers and ICs have reached sub-10 nm features, meanwhile the projected outlook for their fabrication and process technologies indicate the realization of sub-5 and sub-3 nm feature sizes. With the advent and realization of these future trends, the expected increase in defect complexity and frequency necessitates the development of reliable and scalable defect detection algorithms for efficient, and robust RCA and quality monitoring. The advances in ML and DL have subsequently caused immense progress in their application for wafer map defect detection, with the aims to improve model accuracy, cost-efficiency, and production yield.

For data, the wafer map datasets face limitations regarding labeled data availability, class imbalance, and restricted data access. It is known that manual annotation is expensive and time-costly, and as a result, real-world datasets are either small-sized, have a limited range of defects, and/or have a plethora of unlabeled data. Class imbalance persists across many datasets as wafer defects appear at lower frequencies than normal wafer maps; similarly, defect patterns have varied occurrence probabilities. With respect to algorithms that leverage labeled data, the compounding effect of these limitations impose difficulty in achieving robust learning and high-level defect detection, particularly for unknown/rare defect patterns, mixed-type defects, and minor classes. Due to restricted data access, many works directly employ private datasets from semiconductor manufacturing companies. With limited access to real data, synthetic data generation is increasingly appearing in related works to develop more effective models. However, with synthetic data, training generative models to produce realistic wafer maps that are up-to-date on present design standards (i.e., wafer size, IC node size), and similar to real-world defect patterns is difficult and relatively time-costly.

Features are a critical component of ML/DL training. Past works have demonstrated how manual feature generation informed with high-level domain expertise can be advantageous, however, may also lead the derived features to miss hidden/underlying structures. Similarly, with respect to new defects, manually generated features face limitations as important characteristics of the defects may not be known or understood well-enough to generate effective features. The onset of CNNs were prompted by the automated feature extraction capability in which rich, descriptive features can be learned. Similarly, with representation learning, raw data can be used with minimal preprocessing, and can gain high-level discriminatory power for complex patterns; proving that feature representation learning methods can generate more meaningful and effective features for downstream defect pattern classification. However, the capacity of feature representation learning is constrained by model complexity, and whether the model is suited towards learning the complex structure of the data and problem (task) at hand.

For supervised learning methods, although the use of labels can help models achieve improved performance with low computational cost, they are limited by the following: (i) amount of labeled data, (ii) heavily influenced by class label distribution and data splits, and (iii) overfitting. As the acquisition of labels is expensive, limited amounts of high-quality wafer maps are available for training and testing, in which the limited amounts bottleneck classification performance, and highlights limitations in terms of real-world scalability. To tackle this limitation, related works have employed data augmentation techniques to supplement the small-sized datasets, however with the risk of increasing computational costs and generalization error. Similarly, related works implemented specialized modules and deep learning networks to improve learning. With modified modules like the deformable convolutional unit and the usage of specialized loss functions (i.e., contrastive loss, triplet loss), discriminative feature representations were learned. However, these works focused on single-type defects, or considered a limited degree of diversity for the mixed-type defects. In face of new defects and combinations, the performance of these algorithms may decrease with the growth in number of defect classes as class distinctiveness and imbalance may take a toll. As labels are used as the supervisory signal for training, model performance is sensitive to class label distribution, data splits, and class distinctiveness. Class imbalance can induce overfitting on the majority classes, with high misclassification on the minority classes, and may not be able to differentiate between similar defect patterns. This is a critical issue as supervised methods are highly susceptible to overfitting as training is contingent on labels, such that careful considerations should be made for model and training process parameters to prevent overfitting. Relative to defect type, majority of works for supervised algorithms are focused on the detection of single-type defects, and are not suitable for recognizing mixed-type defects, albeit the increasing relevance of mixed-type defects. Multi-label based mixed-type WMDD has limited development and has not be extensively studied for scalability under low resource settings, whereas for multi-class based mixed-type defect detection, much more literature exists. It has noted that additional computational power was required to train the network of binary classifiers, which only considered a limited range of defects, and required large amounts of data for sufficient training. The prominent supervised methods are summarized in Table 6.

As supervised methods face performance limitations due to the amounts of labeled wafer maps, the unsupervised methods demonstrate how the plethora of unlabeled wafer maps can be leveraged. Despite achieving comparable defect detection performance, unsupervised clustering algorithms are sensitive to kernel methods and their respective parameters, and typically have high time complexity, resulting in long run-times. These methods are sensitive to initialization and hyperparameters, indicating the criticality of hyperparameter optimization for performance (Samariya & Thakkar, 2021). Related works have recognized this difficulty of using pre-set parameters (i.e., number of clusters), and in response adapted clustering algorithms with the ability to estimate the number of clusters. However, as these methods involve inference networks, computational complexity increases, attributing long inference speeds which subsequently increases overall run-time. It is important to note that despite the importance of hyperparameter optimization, related works employed simpler optimization frameworks, such as grid search or low-level sensitivity evaluation. Based on reconstruction error, unsupervised pretraining is utilized to improve the initialization of the model weights relative to random initialization, such that training time is faster as the weights are closer to a local optimum. However, in (Alberti et al., 2017), the authors demonstrated how minimizing the reconstruction error for layer-wise training of the autoencoder is not optimal for downstream finetuning for classification tasks as the learned feature representations may not necessarily be meaningful (i.e., an identity function may be learned). The literature for unsupervised pretraining methods demonstrates that representation learning with unlabeled data can be advantageous but needs an effective strategy to learn meaningful feature representations without high computational costs. In Table 7, the performance of the prominent unsupervised clustering algorithms is summarized.

Semi-supervised algorithms address the issues of data availability and ineffective feature learning from supervised and unsupervised methods; demonstrating how the use of both labeled and unlabeled data can achieve improved defect recognition and classification. In particular, the semi-supervised deep generative modelling approach has shown effective latent representation learning, and generative capabilities, but at a relatively high computational cost. It is important to note that with limited amounts of labeled data, model selection is quite important for semi-supervised learning to avoid overfitting and to promote effective representation learning (Kingma et al., 2014). In comparison to supervised and unsupervised methods, the literature for semi-supervised is scarcely developed despite the promising

Table 6 Summary of recognition rates of prominent supervised algorithms for wafer map defect detection

Author	Algorithm	Data	Results
Yu and Lu (2016)	JLNDA	WM-811K (Real)	0.9050
Tello et al. (2018)	RGRN + CNN	Real & Synthetic	Single-type & Mixed-type: 0.8617
Nakazawa and Kulkarni (2018)	CNN	Real	0.9820
Kyeong and Kim (2018)	CNN	Real & Synthetic	Single-type: 0.8700 Mixed-type: 0.9000
Saqlain et al. (2019)	Ensemble	WM-811K (Real)	0.9587
Kong and Ni (2019)	Segmentation + CNN	Real & Synthetic (Mixed-type)	EMR: 0.8775 Accuracy: 0.9575
Kong and Ni (2020b)	Segmentation + CNN	Real (Mixed-type)	0.9010
Wang et al. (2020)	DCN	Mixed WM-38 (Real)	0.9320
Alawieh et al. (2020)	CNN	WM-811K (Real)	0.9400
Zhuang et al. (2020)	DBN	Real	Single-type: 0.9267 Mixed-type: 0.9120
Hyun and Kim (2020)	CNN	Synthetic (Mixed-type)	0.8320
Tsai and Lee (2020a)	CNN	WM-811K (Real)	0.9663
Jin et al. (2020)	CNN-SVM-ECOC	WM-811K (Real)	0.9843
Kim et al. (2021)	SSD	WM-811K (Real)	Single-type: 0.8900 Mixed-type: 0.9200

Table 7 Summary of prominent unsupervised clustering algorithms for wafer map defect detection

Author	Algorithm	Data	Results
Kim et al. (2018)	CPF-iWMM	Real	RI: 0.96 ARI: 0.92 NMI: 0.92
Taha et al. (2018)[a]	DDPfinder	Real & Synthetic	0.9980
Hwang and Kim (2020)	DPGMM	Real	ARI: 0.76 AMI: 0.76

[a] Result reflects the clustering accuracy, which is the algorithm's ability to correctly cluster defect patterns

results; indicating great potential in future developments. In Table 8, the prominent semi-supervised methods are summarized, including the methods that utilized unsupervised pretraining and supervised finetuning.

Enhanced learning strategies were used to boost defect recognition and classification performance. Data augmentation methods utilized image transformations and/or generative models to mitigate class imbalance issues, and subsequently increase data diversity. Although GANs have advanced data generation capabilities, they require substantial computational time to effectively train the generator and discriminator networks. As GAN training involves a trade-off between the generator and discriminator, the models are

Table 8 Summary of prominent semi-supervised algorithms for wafer map defect detection[1]

Author	Algorithm	Data	Results
Yu et al. (2019b)[2]	SCSDAE	WM-811K (Real)	0.9883
Yu (2019)	ESDAE	WM-811K (Real)	0.9703
Hu et al. (2021)[3]	Semi-supervised with contrastive learning	WM-811K (Real)	0.7790
Lee and Kim (2020)[4]	SS-CDGMM	Real & Synthetic	0.9488
Kong and Ni (2020b)	Semi-supervised Ladder Network	Real	0.9020
Kong and Ni (2020a)	SVAE	Real	0.9030
Yu and Liu (2020)	PCACAE	WM-811K (Real)	0.9377
Shon et al. (2021)	CVAE + CNN	WM-811K (Real)	0.9689

[1] Performance results here reflect the overall average recognition rates.
[2] Based on performance with original dataset.
[3] Performance notes the best overall accuracy.
[4] Notes the highest EMR score from the labeled-unlabeled ratios.

susceptible to getting stuck in local minima. For incremental learning strategies, techniques like active learning and pseudo-labeling have demonstrated capability in boosting model performance. However, are susceptible to catastrophic forgetting, and hyperparameter sensitivity (i.e., querying strategy, ratio of original to pseudo-labeled data). With the help of transfer learning, many training processes have been expedited to achieve relatively high accuracy with shorter training times. However, as model complexity, data, and other design choices can impact performance, model selection and hyperparameter tuning need to be carefully considered. For model optimization strategies, RL and BO frameworks are used to bypass the extensive manual searching. These strategies are important in understanding the sensitivities a model may have to input/output, architecture, etc. Although RL imposes extensive training to determine optimal parameters and design, BO provides a more computationally efficient alternative to tuning model hyperparameters. However, it should be noted that for multi-objectives and increasing number of observations, BO frameworks become more computationally complex, which subsequently requires more processing resources.

Challenges and outlook

In this article, we survey the literature of ML and DL applications for wafer map defect recognition and classification, which demonstrated superior performance, as well as great potential and applicability for in-line integration. However, despite the reported successes, many challenges in implementing these methodologies have been identified, including difficulty learning new defects, difficulty differentiating between similar defect patterns, taxing computational loads, and lack of robust detection of complex defects. With respect to the surveyed literature, the following findings emerged as the most prominent challenges in the WMDD field: (i) data availability, (ii) mixed-type defects, and (iii) high computational complexity. The field of WMDD is continuously developed, however, there is limited access to databases that reflect the current design and complexity of wafers and ICs. This is apparent in recent works that utilize the WM-811K dataset, which is most likely outdated in terms of wafer size, IC node size, etc. Similarly, as only private data can properly reflect the present design standards, which has restricted access, the innovation and research for WMDD is slowed. Although simulated data is an option, there is currently a gap in producing realistic, synthetic wafer maps similar to real defect patterns. Majority of existing literature focuses on single-type defects, despite the growing criticality of mixed-type defects. Although mixed-type defect detection algorithms exist, many are limited in terms of labeled data availability, range of defect pattern types, and computational load. Many developments impose a high computational load, which in turn, restricts scalability and potential deployment for real-time implementation, and increases training time and needed processing resources. As this industry will remain competitive and continuously growing, computational complexity should be reduced to be more efficient. These existing challenges impose on implementation, scalability, and adaptability to new state-of-the-art designs and feature sizes.

With the plethora of unlabeled data available, recent developments that leverage ML/DL for self-supervised, and semi-supervised learning indicate potential to surpass supervised learning for efficient feature representation learning, image recognition, and classification. To promote future developments for defect detection, which allows researchers and engineers to validate and test against new designs and feature sizes, consideration towards building a database with real-world defects is needed. Consolidating continuous innovation, growth, and development indicates great promise towards achieving efficient, and robust defect detection. Based on the challenges, and current landscape of this field, the future outlook of WMDD research is summarized as follows:

(1) Handling class imbalance: As many works have focused on tackling the class imbalance issue, it is evident that performance suffers with skewed data distribution. Development for more robust handling of class imbalance is needed, particularly as mixed-type defects become more critical.
(2) Effective unsupervised feature representation learning: As ML/DL, and computer vision applications are increasingly developing self-supervised techniques for image classification and pattern segmentation, these methods should be investigated, especially in face of limited labeled data and the limitations of pretraining via reconstruction loss.
(3) Real-time Monitoring: Majority of developments are offline systems. Consideration of model requirements to meet the conditions needed for real-time monitoring and operation.
(4) Computational Complexity: With respect to real-time monitoring, more efficient and less computational complex algorithms are needed to reduce the burden from training and processing, memory requirements, and scalability limitations.
(5) Model Optimization: Due to the complex parameter-structure-performance relationship, calibrating model selection and the optimal set of parameters is needed. From the existing literature, there is limited exploration and investigation into model optimization and joint hyperparameter tuning.

Acknowledgements The work described in this paper was supported by Natural Sciences and Engineering Research Council of Canada (NSERC under grant RGPIN-217525). The authors are grateful for their support.

Funding This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC), Grant RGPIN-217525.

Appendix

Abbreviation	Term
AC	*Adjacency Clustering*
AdaBalGAN	*Adaptive Balancing Generative Adversarial Network*
AMI	*Adjusted Mutual Information*
ANN	*Artificial Neural Network*
ARI	*Adjusted Rand Index*
BALD	*Bayesian Active Learning by Disagreement*
BO	*Bayesian Optimization*
BB	*Bounding Box*
BPN	*Back Propagation Network*
C2DPCA	*Conditional Two-Dimensional PCA*
CCD	*Charge-Coupled Devices*
CMP	*Chemical Mechanical Process*
CNN	*Convolutional Neural Network*
CPF	*Connected-Path Filtering*
CVAE	*Convolutional Variational Autoencoder*
CZ	*Czochralski*
DBN	*Deep Belief Network*
DBSCAN	*Density-based Spatial Clustering of Applications with Noise*
DCN	*Deformable Convolutional Network*
DCNN	*Deep Convolutional Neural Network*
DDPfinder	*Dominant Defective Patterns Finder*
DFS	*Depth-first Search*
DL	*Deep Learning*
DP	*Dirichlet Process*
ECOC	*Error-Correcting Output Codes*
EMR	*Exact Match Ratio*
ESDAE	*Enhanced Stacked Denoising Autoencoder*
EUV	*Extreme Ultraviolet*
FAM	*Fuzzy ARTMAP*
FD	*Fischer-discriminant*
FZ	*Float-zone*
GAN	*Generative Adversarial Network*
GBM	*Gradient Boosting Machine*
GMM	*Gaussian Mixture Model*
GRN	*Generalized Regression Network*
IC	*Integrated Circuit*
IL	*Incremental Learning*
ILT	*Inverse-lithography Technology*
iWMM	*Infinite Warped Mixture Model*
JLNLDA	*Joint Local and Non-local Linear Discriminant Analysis*
kNN	*k-Nearest Neighbors*
LDA	*Linear Discriminant Analysis*
LLE	*Locally Linear Embedding*
LR	*Logistic Regression*
MDS	*Multi-Dimensional Scaling*
MI	*Mutual Information*
ML	*Machine Learning*
MLP	*Multi-Layer Perceptron*
MPre	*Micro-Precision*
MRe	*Micro-Recall*
NMI	*Normalized Mutual Information*
OPTICS	*Ordering Point to Identify the Cluster Structure*
PCA	*Principal Component Analysis*
PCACAE	*PCA-based Convolutional Autoencoder*
RCA	*Root-cause Analysis*
RF	*Random Forest*
RGRN	*Randomized General Regression Network*
RI	*Rand Index*
RL	*Reinforcement Learning*
RNN	*Recurrent Neural Network*
SAT	*Scanning Acoustic Tomography*
SCSDAE	*Stacked Convolutional Sparse Denoising Autoencoder*

Abbreviation	Term
SDAE	*Stacked Denoising Autoencoder*
SEM	*Scanning Electron Microscopy*
SS-CDGMM	*Semi-supervised Convolutional Deep Generative Multiple Models*
SSD	*Single Shot Detector*
SVAE	*Semi-supervised Variational Autoencoder*
SVC	*Support Vector Clustering*
SVE	*Soft Voting Ensemble*
SVM	*Support Vector Machine*
t-SNE	*t-distributed Stochastic Neighbor Embedding*
TTV	*Total Thickness Variation*
UV	*Ultraviolet*
VAE	*Variational Autoencoder*
WBM	*Wafer Bin Map*
WM	*Wafer Map*
WMDD	*Wafer Map Defect Detection*

References

Adly, F., Alhussein, O., Yoo, P. D., Al-Hammadi, Y., Taha, K., Muhaidat, S., Jeong, Y.-S., Lee, U., & Ismail, M. (2015a). Simplified subspaced regression network for identification of defect patterns in semiconductor wafer maps. *IEEE Transactions on Industrial Informatics, 11*(6), 1267–1276. https://doi.org/10.1109/TII.2015.2481719

Adly, F., Yoo, P., Muhaidat, S., Al-Hammadi, Y., Lee, U., & Ismail, M. (2015b). Randomized general regression network for identification of defect patterns in semiconductor wafer maps. *IEEE Transactions on Semiconductor Manufacturing, 28*(2), 145–152. https://doi.org/10.1109/tsm.2015.2405252

Airaksinen, V.-M. (2015). Silicon wafer and thin film measurements. In M. Tilli, T. Motooka, V.-M. Airaksinen, S. Franssila, M. Paulasto-Kröckel , & V. Lindroos (Eds.), *Handbook of Silicon Based MEMS Materials and Technologies* (2nd Ed., pp. 381–390). https://doi.org/10.1016/B978-0-323-29965-7.00015-4

Alawieh, M. B., Boning, D., & Pan, D. Z. (2020). Wafer map defect patterns classification using deep selective learning. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. https://doi.org/10.1109/dac18072.2020.9218580

Alawieh, M. B., Wang, F., & Li, X. (2018). Identifying wafer-level systematic failure patterns via unsupervised learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37*(4), 832–844. https://doi.org/10.1109/TCAD.2017.2729469

Alberti, M., Seuret, M., Ingold, R., & Liwicki, M. (2017, December 17). *A Pitfall of Unsupervised Pre-Training*. arXiv.org. Retrieved from https://arxiv.org/abs/1712.01655.

Baker, B., Raskar, R., Naik, N., & Gupta, O. (2017). Designing Neural Network Architectures using Reinforcement Learning. In *Proc. of ICLR 2017*. Retrieved from https://arxiv.org/abs/1611.02167.

Batool, U., Shapiai, M. I., Fauzi, H., & Fong, J. X. (2020). Convolutional neural network for imbalanced data classification of silicon wafer defects. In *2020 16th IEEE International Colloquium on Signal Processing Its Applications (CSPA)*, 230–235. https://doi.org/10.1109/CSPA48992.2020.9068669

Bella, R. D., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G. (2019, September). Wafer defect map classification using sparse convolutional networks. In *International Conference on Image Analysis and Processing* (pp. 125–136). Springer, Cham.

Bello, I., Zoph, B., Vasudevan, V., & Le, Q. V. (2017). Neural Optimizer Search with Reinforcement Learning. In *Proceedings of 34th International Conference on Machine Learning* (pp. 459–468). Sydney. Retrieved from https://arxiv.org/abs/1709.07417.

Banchhor, C., & Srinivasu, N. (2021). Analysis of Bayesian optimization algorithms for big data classification based on Map Reduce framework. *Journal of Big Data, 8*(1), 81. https://doi.org/10.1186/s40537-021-00464-4

Byun, Y., & Baek, J. G. (2020). Mixed pattern recognition methodology on wafer maps with pre-trained convolutional neural networks. In A. Rocha, L. Steels, & J. van den Herik (Eds.), *ICAART 2020 - Proceedings of the 12th International Conference on Agents and Artificial Intelligence* (pp. 974–979). (ICAART 2020—Proceedings of the 12th International Conference on Agents and Artificial Intelligence; Vol. 2). SciTePress.

Chang, C.-W., Chao, T.-M., Horng, J.-T., Lu, C.-F., & Yeh, R.-H. (2012). Development pattern recognition model for the classification of circuit probe wafer maps on semiconductors. *IEEE Transactions on Components, Packaging and Manufacturing Technology, 2*(12), 2089–2097. https://doi.org/10.1109/TCPMT.2012.2215327

Chen, H.-C. (2020). Automated detection and classification of defective and abnormal dies in wafer images. *Applied Sciences, 10*(10), 3423. https://doi.org/10.3390/app10103423

Chen, S.-H., Kang, C.-H., & Perng, D.-B. (2020a). Detecting and measuring defects in wafer die using GAN and YOLOv3. *Applied Sciences, 10*(23), 8725. https://doi.org/10.3390/app10238725

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020b). *A Simple Framework for Contrastive Learning of Visual Representations*.

Chen, F.-L., & Liu, S.-F. (2000). A neural-network approach to recognize defect spatial pattern in semiconductor fabrication. *IEEE Transactions on Semiconductor Manufacturing, 13*(3), 366–373. https://doi.org/10.1109/66.857947

Cheon, S., Lee, H., Kim, C. O., & Lee, S. H. (2019). Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. *IEEE Transactions on Semiconductor Manufacturing, 32*(2), 163–170. https://doi.org/10.1109/tsm.2019.2902657

Chien, C.-F., Hsu, S.-C., & Chen, Y.-J. (2013). A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence. *International Journal of Production Research, 51*(8), 2324–2338. https://doi.org/10.1080/00207543.2012.737943

Chien, J.-C., Wu, M.-T., & Lee, J.-D. (2020). Inspection and classification of semiconductor wafer surface defects using CNN deep learning networks. *Applied Sciences, 10*(15), 5340. https://doi.org/10.3390/app10155340

Choi, G., Kim, S.-H., Ha, C., & Bae, S. J. (2012). Multi-step ART1 algorithm for recognition of defect patterns on semiconductor wafers. *International Journal of Production Research, 50*(12), 3274–3287. https://doi.org/10.1080/00207543.2011.574502

Cuevas, A., & Sinton, R. A. (2018). Chapter III-1-A - Characterization and Diagnosis of Silicon Wafers, Ingots, and Solar Cells. In D. Macdonald & S. A. Kalogirou (Eds.), *McEvoy's Handbook of Photovoltaics* (3rd Ed., pp. 1119–1154). Essay, Academic Press.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv.2017.89

Devika, B., & George, N. (2019). Convolutional neural network for semiconductor wafer defect detection. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1–6). https://doi.org/10.1109/ICCCNT45670.2019.8944584

di Palma, F., de Nicolao, G., Miraglia, G., Pasquinetti, E., & Piccinini, F. (2005). Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing. *Pattern Recognition Letters, 26*(12), 1857–1865. https://doi.org/10.1016/j.patrec.2005.03.007

Du, D.-Y., & Shi, Z. (2020). A wafer map defect pattern classification model based on deep convolutional neural network. In *2020 IEEE 15th International Conference on Solid-State Integrated Circuit Technology (ICSICT)* (pp. 1–3). https://doi.org/10.1109/ICSICT49897.2020.9278021

Ebayyeh, A. A., & Mousavi, A. (2020). A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry. *IEEE Access, 8*, 183192–183271. https://doi.org/10.1109/access.2020.3029127

Ezzat, A. A., Liu, S., Hochbaum, D. S., & Ding, Y. (2021). A graph-theoretic approach for spatial filtering and its impact on mixed-type spatial pattern recognition in wafer bin maps. *IEEE Transactions on Semiconductor Manufacturing, 34*(2), 194–206. https://doi.org/10.1109/tsm.2021.3062943

Faaeq, A., Guruler, H., & Peker, M. (2018). Image classification using manifold learning based non-linear dimensionality reduction. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*. https://doi.org/10.1109/siu.2018.8404441

Fan, M., Wang, Q., & van der Waal, B. (2016). Wafer defect patterns recognition based on OPTICS and multi-label classification. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 912–915. https://doi.org/10.1109/IMCEC.2016.7867343

Hasan, R. M., & Luo, X. (2018). Promising lithography techniques for next-generation logic devices. *Nanomanufacturing and Metrology, 1*(2), 67–81. https://doi.org/10.1007/s41871-018-0016-9

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. B. (2019). Momentum Contrast for Unsupervised Visual Representation Learning. *CoRR, abs/1911.05722*. http://arxiv.org/abs/1911.05722

Hsu, S.-C., & Chien, C.-F. (2007). Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing. *International Journal of Production Economics, 107*(1), 88–103. https://doi.org/10.1016/j.ijpe.2006.05.015

Hsu, C.-Y., Chen, W.-J., & Chien, J.-C. (2020). Similarity matching of wafer bin maps for manufacturing intelligence to empower Industry 3.5 for semiconductor manufacturing. *Computers & Industrial Engineering, 142*, 106–358. https://doi.org/10.1016/j.cie.2020.106358

Hu, M. (1962). Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory, 8*(2), 179–187. https://doi.org/10.1109/tit.1962.1057692

Hu, H., He, C., & Li, P. (2021). Semi-supervised wafer map pattern recognition using domain-specific data augmentation and contrastive learning. *IEEE International Test Conference (ITC), 2021*, 113–122. https://doi.org/10.1109/ITC50571.2021.00019

Huang, C.-J. (2007). Clustered defect detection of high quality chips using self-supervised multilayer perceptron. *Expert Systems with Applications, 33*(4), 996–1003. https://doi.org/10.1016/j.eswa.2006.07.011

Huang, C.-J., Chen, Y.-J., Wu, C.-F., & Huang, Y.-A. (2009). Application of neural networks and genetic algorithms to the screening for high quality chips. *Applied Soft Computing, 9*(2), 824–832. https://doi.org/10.1016/j.asoc.2008.10.002

Hwang, J., & Kim, H. (2020). Variational deep clustering of wafer map patterns. *IEEE Transactions on Semiconductor Manufacturing, 33*(3), 466–475. https://doi.org/10.1109/tsm.2020.3004483

Hyun, Y., & Kim, H. (2020). Memory-augmented convolutional neural networks with triplet loss for imbalanced wafer defect pattern classification. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 622–634. https://doi.org/10.1109/tsm.2020.3010984

Ishida, T., Nitta, I., Fukuda, D., & Kanazawa, Y. (2019). Deep learning-based wafer-map failure pattern recognition framework. In *20th International Symposium on Quality Electronic Design (ISQED)*. https://doi.org/10.1109/isqed.2019.8697407

Iwata, T., Duvenaud, D., & Ghahramani, Z. (2013, March 21). *Warped Mixtures for Nonparametric Cluster Shapes*. arXiv.org. Retrieved from https://arxiv.org/abs/1206.1846.

Jang, J., Seo, M., & Kim, C. O. (2020). Support weighted ensemble model for open set recognition of wafer map defects. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 635–643. https://doi.org/10.1109/tsm.2020.3012183

Ji, Y. S., & Lee, J.-H. (2020). Using GAN to improve CNN performance of wafer map defect type classification: Yield enhancement. In 2020 31st Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC). https://doi.org/10.1109/asmc49169.2020.9185193

Jin, C. H., Kim, H.-J., Piao, Y., Li, M., & Piao, M. (2020). Wafer map defect pattern classification based on convolutional neural network features and error-correcting output codes. *Journal of Intelligent Manufacturing, 31*(8), 1861–1875. https://doi.org/10.1007/s10845-020-01540-x

Jin, C. H., Na, H. J., Piao, M., Pok, G., & Ryu, K. H. (2019). A novel DBSCAN-based defect pattern detection and classification framework for wafer bin map. *IEEE Transactions on Semiconductor Manufacturing, 32*(3), 286–292. https://doi.org/10.1109/tsm.2019.2916835

Kang, S. (2020). Rotation-invariant wafer map pattern classification with convolutional neural networks. *IEEE Access, 8*, 170650–170658. https://doi.org/10.1109/access.2020.3024603

Kang, H., & Kang, S. (2021). A stacking ensemble classifier with handcrafted and convolutional features for wafer map pattern classification. *Computers in Industry, 129*, 103450. https://doi.org/10.1016/j.compind.2021.103450

Khastavaneh H., & Ebrahimpour-Komleh H. (2020) Representation learning techniques: An overview. In Bohlouli, M., Sadeghi Bigham, B., Narimani, Z., Vasighi, M., & Ansari, E. (Eds.), Data Science: From Research to Application. CiDaS 2019. Lecture Notes on Data Engineering and Communications Technologies, Vol. 45. Springer, Cham. https://doi.org/10.1007/978-3-030-37309-2_8

Kim, Y., Cho, D., & Lee, J.-H. (2020a). Wafer map classifier using deep learning for detecting out-of-distribution failure patterns. *IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA), 2020*, 1–5. https://doi.org/10.1109/IPFA49335.2020.9260877

Kim, B., Jeong, Y.-S., Tong, S. H., & Jeong, M. K. (2020b). A generalised uncertain decision tree for defect classification of multiple wafer maps. *International Journal of Production Research, 58*(9), 2805–2821. https://doi.org/10.1080/00207543.2019.1637035

Kim, J., Lee, Y., & Kim, H. (2018). Detection and clustering of mixed-type defect patterns in wafer bin maps. *IISE Transactions, 50*(2), 99–111. https://doi.org/10.1080/24725854.2017.1386337

Kim, T. S., Lee, J. W., Lee, W. K., & Sohn, S. Y. (2021). Novel method for detection of mixed-type defect patterns in wafer maps based on a single shot detector algorithm. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01755-6

Kim, S., & Oh, I. S. (2017). Automatic Defect Detection from SEM Images of Wafers using Component Tree. *JSTS Journal of Semiconductor Technology and Science, 17*(1), 86–93. https://doi.org/10.5573/jsts.2017.17.1.086

Kingma, D. P., Rezende, D. J., Mohamed, S., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems* (Vol. 4, pp. 3581–3589).

Kong, Y., & Ni, D. (2018). Semi-supervised classification of wafer map based on ladder network. In *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*. https://doi.org/10.1109/icsict.2018.8564982

Kong, Y., & Ni, D. (2019). Recognition and location of mixed-type patterns in wafer bin maps. In *2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*. https://doi.org/10.1109/smile45626.2019.8965309

Kong, Y., & Ni, D. (2020a). A semi-supervised and incremental modeling framework for wafer map classification. *IEEE Transactions on Semiconductor Manufacturing, 33*(1), 62–71. https://doi.org/10.1109/tsm.2020.2964581

Kong, Y., & Ni, D. (2020b). Qualitative and quantitative analysis of multi-pattern wafer bin maps. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 578–586. https://doi.org/10.1109/tsm.2020.3022431

Kyeong, K., & Kim, H. (2018). Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. *IEEE Transactions on Semiconductor Manufacturing, 31*(3), 395–402. https://doi.org/10.1109/tsm.2018.2841416

Lee, H., & Kim, H. (2020). Semi-supervised multi-label learning for classification of wafer bin maps with mixed-type defect patterns. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 653–662. https://doi.org/10.1109/tsm.2020.3027431

Li, K., Liao, P., Cheng, K., Chen, L., Wang, S., Huang, A., et al. (2021). Hidden wafer scratch defects projection for diagnosis and quality enhancement. *IEEE Transactions on Semiconductor Manufacturing, 34*(1), 9–16. https://doi.org/10.1109/tsm.2020.3040998

Li, T.-S., & Huang, C.-L. (2009). Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing. *Expert Systems with Applications, 36*(1), 374–385. https://doi.org/10.1016/j.eswa.2007.09.023

Liao, C.-S., Hsieh, T.-J., Huang, Y.-S., & Chien, C.-F. (2014). Similarity searching for defective wafer bin maps in semiconductor manufacturing. *IEEE Transactions on Automation Science and Engineering, 11*(3), 953–960. https://doi.org/10.1109/TASE.2013.2277603

Liu, C.-W., & Chien, C.-F. (2013). An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing. *Engineering Applications of Artificial Intelligence, 26*(5–6), 1479–1486. https://doi.org/10.1016/j.engappai.2012.11.009

Luo, Y., Yin, L., Bai, W., & Mao, K. (2020). An appraisal of incremental learning methods. *Entropy, 22*(11), 1190. https://doi.org/10.3390/e22111190

Maksim, K., Kirill, B., Eduard, Z., Nikita, G., Aleksandr, B., Arina, L., Vladislav, S., Daniil, M., & Nikolay, K. (2019). Classification of wafer maps defect based on deep learning methods with small amount of data. *International Conference on Engineering and Telecommunication (EnT), 2019*, 1–5. https://doi.org/10.1109/EnT47717.2019.9030550

Mohanaiah, P., Sathyanarayana, P., & GuruKumar, L. (2013). Image texture feature extraction using GLCM approach. *International Journal of Scientific and Research Publications, 3*(5), 1–5.

Nakazawa, T., & Kulkarni, D. V. (2018). Wafer map defect pattern classification and image retrieval using convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing, 31*(2), 309–314. https://doi.org/10.1109/tsm.2018.2795466

Northcutt, C., Jiang, L., & Chuang, I. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research, 70*, 1373–1411. https://doi.org/10.1613/jair.1.12125

Ooi, M.P.-L., Sok, H. K., Kuang, Y. C., Demidenko, S., & Chan, C. (2013). Defect cluster recognition system for fabricated semiconductor wafers. *Engineering Applications of Artificial Intelligence, 26*(3), 1029–1043. https://doi.org/10.1016/j.engappai.2012.03.016

Park, S., Jang, J., & Kim, C. O. (2020). Discriminative feature learning and cluster-based defect label reconstruction for reducing uncertainty in wafer bin map label. *Journal of Intelligent Manufacturing, 32*(1), 251–263. https://doi.org/10.1007/s10845-020-01571-4

Parsa, M., Mitchell, J. P., Schuman, C. D., Patton, R. M., Potok, T. E., & Roy, K. (2020). Bayesian multi-objective hyperparameter optimization for accurate, fast, and efficient neural network accelerator design. *Frontiers in Neuroscience*. https://doi.org/10.3389/fnins.2020.00667

Patel, D. V., Bonam, R., & Oberai, A. A. (2020). Deep learning-based detection, classification, and localization of defects in semiconductor processes. *Journal of Micro/nanolithography, MEMS, and MOEMS, 19*(02), 1. https://doi.org/10.1117/1.jmm.19.2.024801

Patel, S., Sihmar, S., & Jatain, A. (2015). A study of hierarchical clustering algorithms. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 537–541).

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. CoRR, abs/1712.04621. http://arxiv.org/abs/1712.04621

Piao, M., Jin, C. H., Lee, J. Y., & Byun, J.-Y. (2018). Decision tree ensemble-based wafer map failure pattern recognition based on radon transform-based features. *IEEE Transactions on Semiconductor Manufacturing, 31*(2), 250–257. https://doi.org/10.1109/TSM.2018.2806931

Pleschberger, M., Scheiber, M., & Schrunner, S. (2019). Simulated analog wafer test data for pattern recognition. Zenodo. https://doi.org/10.5281/zenodo.2542504

Preil, M. E. (2016). Patterning challenges in the sub-10 nm era. *Optical Microlithography XXIX, 10*(1117/12), 2222256.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Chen, X., & Wang, X. (2020, August 30). *A survey of deep active learning*. arXiv.org. Retrieved September 22, 2021, from https://arxiv.org/abs/2009.00236.

Ruthotto, L., & Haber, E. (2021). An introduction to deep generative modeling. *GAMM-Mitteilungen*. https://doi.org/10.1002/gamm.202100008

Samariya, D., & Thakkar, A. (2021). A comprehensive survey of anomaly detection algorithms. *Annals of Data Science*. https://doi.org/10.1007/s40745-021-00362-9

Santos, A. M., & Canuto, A. M. P. (2012). Using semi-supervised learning in multi-label classification problems. In *The 2012 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). https://doi.org/10.1109/IJCNN.2012.6252800

Saqlain, M., Abbas, Q., & Lee, J. Y. (2020). A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing, 33*(3), 436–444. https://doi.org/10.1109/tsm.2020.2994357

Saqlain, M., Jargalsaikhan, B., & Lee, J. Y. (2019). A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 32*(2), 171–182. https://doi.org/10.1109/tsm.2019.2904306

Settles, B. (2009). (rep.). *Active Learning Literature Survey*. Madison, WI.

Shawon, A., Faruk, M. O., Habib, M. B., & Khan, A. M. (2019). Silicon wafer map defect classification using deep convolutional neural network with data augmentation. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. https://doi.org/10.1109/iccc47050.2019.9064029

Shen, Z., & Yu, J. (2019). Wafer map defect recognition based on deep transfer learning. In *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. https://doi.org/10.1109/ieem44572.2019.8978568

Shi, X., Yan, Y., Zhou, T., Yu, X., Li, C., Chen, S., & Zhao, Y. (2020). Fast and Accurate Machine Learning Inverse Lithography Using Physics Based Feature Maps and Specially Designed DCNN. In *2020 International Workshop on Advanced Patterning Solutions (IWAPS)*. https://doi.org/10.1109/iwaps51164.2020.9286814

Shi, X., Zhao, Y., Cheng, S., Li, M., Yuan, W., Yao, L., Zhao, W., Xiao, Y., Kang, X., & Li, A. (2019). Optimal feature vector design for computational lithography. *Optical Microlithography XXXII, 10*(1117/12), 2515446.

Shim, J., Kang, S., & Cho, S. (2020). Active learning of convolutional neural network for cost-effective wafer map pattern classification. *IEEE Transactions on Semiconductor Manufacturing, 33*(2), 258–266. https://doi.org/10.1109/tsm.2020.2974867

Shon, H. S., Batbaatar, E., Cho, W.-S., & Choi, S. G. (2021). Unsupervised pre-training of imbalanced data for identification of wafer map defect patterns. *IEEE Access, 9*, 52352–52363. https://doi.org/10.1109/access.2021.3068378

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, Md. M. A., Prabhat, & Adams, R. P. (2015). Scalable Bayesian Optimization Using Deep Neural Networks. *ArXiv E-Prints*, arXiv:1502.05700.

Taha, K., Salah, K., & Yoo, P. D. (2018). Clustering the dominant defective patterns in semiconductor wafer maps. *IEEE Transactions on Semiconductor Manufacturing, 31*(1), 156–165. https://doi.org/10.1109/TSM.2017.2768323

Tello, G., Al-Jarrah, O., Yoo, P., Al-Hammadi, Y., Muhaidat, S., & Lee, U. (2018). Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes. *IEEE Transactions on Semiconductor Manufacturing, 31*(2), 315–322. https://doi.org/10.1109/tsm.2018.2825482

Tsai, T.-H., & Lee, Y.-C. (2020a). Wafer Map Defect Classification with Depthwise Separable Convolutions. In *2020a IEEE International Conference on Consumer Electronics (ICCE)*. https://doi.org/10.1109/icce46568.2020a.9043041

Tsai, T.-H., & Lee, Y.-C. (2020b). A light-weight neural network for wafer map classification based on data augmentation. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 663–672. https://doi.org/10.1109/TSM.2020.3013004

Ungredda, J., & Branke, J. (2021). Bayesian optimisation for constrained problems. *CoRR*, *abs/2105.13245*. https://arxiv.org/abs/2105.13245

Wang, C.-H. (2009). Separation of composite defect patterns on wafer bin map using support vector clustering. *Expert Systems with Applications, 36*(2, Part 1), 2554–2561. https://doi.org/10.1016/j.eswa.2008.01.057

Wang, C.-H. (2008). Recognition of semiconductor defect patterns using spatial filtering and spectral clustering. *Expert Systems with Applications, 34*(3), 1914–1923. https://doi.org/10.1016/j.eswa.2007.02.014

Wang, R., & Chen, N. (2019). Wafer map defect pattern recognition using rotation-invariant features. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 596–604. https://doi.org/10.1109/TSM.2019.2944181

Wang, C.-H., Kuo, W., & Bensmail, H. (2006). Detection and classification of defect patterns on semiconductor wafers. *IIE Transactions, 38*(12), 1059–1068. https://doi.org/10.1080/07408170600733236

Wang, J., Xu, C., Yang, Z., Zhang, J., & Li, X. (2020). Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. *IEEE Transactions on Semiconductor Manufacturing, 33*(4), 587–596. https://doi.org/10.1109/tsm.2020.3020985

Wang, J., Yang, Z., Zhang, J., Zhang, Q., & Chien, W.-T.K. (2019). AdaBalGAN: An improved generative adversarial network with imbalanced learning for wafer defective pattern recognition. *IEEE Transactions on Semiconductor Manufacturing, 32*(3), 310–319. https://doi.org/10.1109/tsm.2019.2925361

Wang, W., Huang, Y., Wang, Y., & Wang, L. (2014). Generalized autoencoder: A neural network framework for dimensionality reduction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. https://doi.org/10.1109/cvprw.2014.79

Wang, Y., & Ni, D. (2019). Multi-bin Wafer Maps Defect Patterns Classification. In *2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE)*. https://doi.org/10.1109/smile45626.2019.8965299

Wen, G., Gao, Z., Cai, Q., Wang, Y., & Mei, S. (2020). A novel method based on deep convolutional neural networks for wafer semiconductor surface defect inspection. *IEEE Transactions on Instrumentation and Measurement, 69*(12), 9668–9680. https://doi.org/10.1109/tim.2020.3007292

White, K. P., Kundu, B., & Mastrangelo, C. M. (2008). Classification of defect clusters on semiconductor wafers via the hough transformation. *IEEE Transactions on Semiconductor Manufacturing, 21*(2), 272–278. https://doi.org/10.1109/tsm.2008.2000269

Wu, M.-J., Jang, J.-S.R., & Chen, J.-L. (2015). Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Transactions on Semiconductor Manufacturing, 28*(1), 1–12. https://doi.org/10.1109/tsm.2014.2364237

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science, 2*(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1

Yu, J. (2019). Enhanced stacked denoising autoencoder-based feature learning for recognition of wafer map defects. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 613–624. https://doi.org/10.1109/tsm.2019.2940334

Yu, J., & Liu, J. (2020). Two-dimensional principal component analysis-based convolutional autoencoder for wafer map defect detection. *IEEE Transactions on Industrial Electronics, 68*(9), 8789–8797. https://doi.org/10.1109/tie.2020.3013492

Yu, J., & Lu, X. (2016). Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. *IEEE Transactions on Semiconductor Manufacturing, 29*(1), 33–43. https://doi.org/10.1109/tsm.2015.2497264

Yu, N., Xu, Q., & Wang, H. (2019a). Wafer defect pattern recognition and analysis based on convolutional neural network. *IEEE Transactions on Semiconductor Manufacturing, 32*(4), 566–573. https://doi.org/10.1109/TSM.2019.2937793

Yu, J., Zheng, X., & Liu, J. (2019b). Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map. *Computers in Industry, 109*, 121–133. https://doi.org/10.1016/j.compind.2019.04.015

Yuan, T., Bae, S. J., & Park, J. I. (2010). Bayesian spatial defect pattern recognition in semiconductor fabrication using support vector clustering. *The International Journal of Advanced Manufacturing Technology, 51*(5), 671–683. https://doi.org/10.1007/s00170-010-2647-x

Zhong, G., Wang, L., Ling, X., & Dong, J. (2016). An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science, 2*(4), 265–278. https://doi.org/10.1016/j.jfds.2017.05.001

Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable ConvNets V2: More Deformable, Better Results. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2019.00953

Zhuang, J., Mao, G., Wang, Y., Chen, X., & Wei, Z. (2020). A neural-network approach to better diagnosis of defect pattern in wafer bin map. *China Semiconductor Technology International Conference (CSTIC), 2020*, 1–3. https://doi.org/10.1109/CSTIC49141.2020.9282438

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018, April 11). *Learning transferable architectures for scalable image recognition*. arXiv.org. Retrieved September 22, 2021, from https://arxiv.org/abs/1707.07012.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.