

패턴 불균형에 강건한 자가 지도학습을 활용한 웨이퍼 불량 패턴 클러스터링 방법 제안

최이수 · 윤주호 · 김병훈[†]

한양대학교 산업경영공학과

A New Wafer Defect Pattern Clustering Method using a Self Supervised Learning Robust to Pattern Imbalance

Leesu Choi · Juho Yoon · Byunghoon Kim

Department of Industrial and Management Engineering, Hanyang University

This study proposes a wafer defect pattern clustering model that can recognize defect patterns without the class label of the defect patterns. In the first step, noise defects are removed from each wafer bin map (WBM) image using the Depth-First Search (DFS) algorithm to clarify the defect pattern. Next, the defect patterns are clustered using the Dirichlet process, and the clustering results are adjusted by tuning the extracted features based on self-supervised learning. By employing a weighted cross-entropy loss that considers the cluster size, the model becomes robust to the imbalance of cluster sizes during the fine-tuning process. The proposed method can facilitate the identification and resolution of the causes of defects that occur during semiconductor processing.

Keywords: Defect pattern clustering, Self-supervised learning, Semiconductor processing, Wafer bin map

1. 서론

1.1 웨이퍼 불량 유형 분류 모델 개발의 필요성

반도체는 전류가 흐르는 도체와 전류가 흐르지 않는 부도체의 중간 성질을 가지며, 얇은 원판인 웨이퍼에서 8개의 메인 공정(산화, 포토, 에칭, 박막 증착, 금속 배선, Electrical Die Sorting(EDS), 패키징 등)을 거쳐 제조된다(Yoo *et al.*, 2015). 이후 EDS 공정을 통해 웨이퍼의 모든 반도체 칩에 대해 전기적 검사를 수행하고, Wafer Bin Map(WBM)을 통해 불량 유형을 파악할 수 있다.(Liao *et al.*, 2013).

대표적인 WBM의 불량 유형은 <Figure 1>과 같으며. 같은

불량 패턴을 가지는 WBM은 같은 공정에서 불량이 발생할 가능성이 높다(Yu and Lu, 2016). 예를 들어 원형(Center) 불량인 경우 박막 증착 공정과 연관되어 있으며 Edge Ring 불량인 경우 식각 공정, 그리고 스크래치의 경우 웨이퍼 핸들링 과정에서 불량이 발생했을 가능성이 높다(Kang, 2020). 따라서 엔지니어들은 WBM 불량 패턴을 파악하여 불량 원인이 되는 공정을 빠르게 파악하고 해결하는 것이 중요하다.

하지만 대부분의 기업은 엔지니어의 경험과 육안에 의존하여 불량 패턴을 파악하고 있으며, 대규모의 반도체 공정에서 웨이퍼를 일일이 파악하는 것은 시간과 비용 측면에서 매우 소모적이다(Liu *et al.*, 2002). 또한 엔지니어의 사전 지식에 따라 동일한 불량 패턴도 다른 유형의 불량 패턴으로 파악할 수

이 논문은 2023년도 산업통상자원부 및 한국산업기술진흥원의 재원으로 한국전자기술연구원의 지원을받아 수행된 인력양성사업임(P0008691). 이 논문은 2022년도 한국데이터마이닝학회 SAS 논문 경진대회에서 장려상을 수상한 논문을 확장하여 작성한 논문임. 이 논문은 한국연구재단의 연구비 지원을 받아 수행됨(No. NRF-2019R1F1A1042307).

[†] 연락저자 : 김병훈 교수, 15588 경기도 안산시 상록구 한양대학교 55 5공학관 510호, Tel : 031-400-5269, E-mail : byungkim@hanyang.ac.kr
2023년 4월 11일 접수; 2023년 6월 19일 수정본 접수; 2023년 6월 20일 게재 확정.

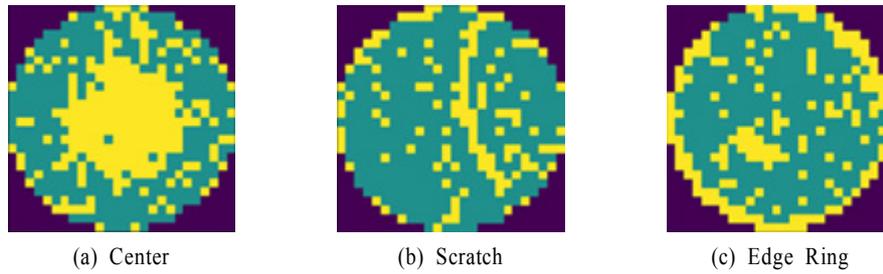


Figure 1. Example of WBM Defect Patterns

있는 문제점이 있어, 패턴 인식을 통해 웨이퍼 불량 유형을 분류하는 모델을 개발하는 것이 중요하다(Chen *et al.*, 2020).

1.2 문제 정의

과거 WBM 관련 연구는 대부분 불량 패턴 분류(Classification)에 초점을 두었으며, 이는 지도학습(Supervised Learning)의 한 형태로서 모든 WBM 데이터에 대해 불량 패턴 라벨링이 필요한 단점이 있다(Yu *et al.*, 2019). 그러나 반도체 공정의 특성 상 모든 데이터에 정확한 불량 패턴을 라벨링 하는 것은 매우 어렵고, 많은 시간과 비용이 소모된다. 또한, 사전에 불량 유형을 미리 정해줌으로써 새로운 불량 패턴은 탐지하지 못한다는 한계가 있다(Baek *et al.*, 2020).

또한 WBM 불량 패턴 클러스터링 관련 연구들의 경우 데이터 불균형을 고려하지 못하였다(Kim and Kang, 2021). 즉, 웨이퍼 불균형을 고려하지 않으면 비율이 높은 불량 패턴을 중심으로 학습이 진행되어 비율이 낮은 불량 패턴은 검출되지 않는 한계가 있다. 특히 얇고 긴 스크래치 패턴은 지도학습과 비지도학습 연구 모두에서 제대로 탐지하지 못하고 있다. 이는 반도체 제조 공정 내 웨이퍼 핸들링 오류로 인한 불량은 검출하지 못할 수도 있음을 시사한다(Alawieh *et al.*, 2020).

따라서 라벨링 되지 않은 웨이퍼 불량 패턴을 클러스터링할 때, 사이즈의 불균형을 고려하여 모델을 구현하고, 동시에 스크래치 패턴의 탐지율을 향상시켜 정확한 반도체 공정 내 불량 원인을 파악하는 것이 필요하다.

1.3 기여점

본 연구에서는 라벨이 필요하지 않은 클러스터링 알고리즘을 사용하여 엔지니어의 개입 없이 불량 패턴을 자동으로 클러스터링하는 모델을 제안한다. 이때 불량 패턴 개수를 자동으로 정해줌으로써 새로운 패턴과 이상치에도 강건하고, 깊이 우선 알고리즘(Depth First Search, DFS)을 노이즈 제거에 적용하여 스크래치 검출 성능을 향상시켰다. 또한 학습 시 데이터의 클러스터별 가중치를 사용하여 데이터 불균형 문제를 해결하였으며, 자가지도학습(Semi-Supervised Learning)을 이용하여 잠재 벡터를 최적화함으로써 모델의 강건성을 높였다. 제

안한 방법을 통해 WBM 불량 패턴의 실시간 파악이 가능하며, 반도체 제조 공정 내에서 불량 원인의 파악을 자동화하여 공정 결함을 조기에 발견하고 해결함으로써 반도체 수율 향상과 비용 절감 효과를 기대할 수 있다(Ko *et al.*, 2020).

본 연구는 논문의 다음과 같이 구성된다. 제2장에서는 WBM 불량 패턴과 관련된 연구를 소개하며, 제3장에서는 제안 방법론을 설명한다. 제4장에서는 실험 설계와 평가 방법을 제시하고, 마지막으로 제5장에서는 결론과 향후 연구 방향에 대해 논의한다.

2. 관련 연구

2.1 지도학습을 통한 패턴 분류

Nakazawa and Kulkarn(2018)는 시뮬레이션을 통해 22가지의 불량 패턴을 생성한 후 Convolutional Neural Network(CNN)을 이용하여 분류 모델을 구현하였다. 하지만 실제 WBM을 이용하여 평가한 결과 Curved Scratch 및 Line Scratch 유형의 정확도가 다소 떨어졌다. CNN 기반의 딥러닝 모델은 데이터에 정확한 라벨링을 필요로 하며, 새로운 패턴과 이상치를 탐지할 수 없다는 한계가 있고, 스크래치 패턴과 같은 일부 불량 유형에서는 성능이 다소 떨어진다는 한계가 있다.

Kahng and Kim(2021)은 자가지도학습을 이용하여 라벨이 없는 데이터에도 라벨링을 적용하는 연구를 수행하였다. 각각 1%, 5%, 10%, 25%, 50% 라벨링 된 데이터를 증강(Augmentation)하여, 생성된 이미지와 기존 이미지의 유사도 기반 학습을 통해 자가지도학습을 진행하여 라벨이 없는 데이터에도 라벨링을 적용하였다. 그러나 불량 패턴의 유형을 사전에 정의해야 하고, 라벨링이 필요한 자가지도학습 방법의 한계가 있다.

Song and Kim(2022)은 사전에 정의한 불량 패턴뿐만 아니라 신규 불량 패턴을 탐지하는 연구를 수행하였다. 사전에 정의한 불량 패턴들에 대한 분포인 In-distribution을 학습시킨 후 Deep-MCDD(Multi Class Data Description) 모델을 사용하여 OOD(Out of Distributions)를 탐지하여 신규 불량 패턴을 검출하였다. 이를 통해 사전에 정의되어 있는 불량 패턴의 성능을 높이는 방식으로만 학습이 진행되어 신규 불량 패턴을 탐지하

지 못한다는 기존 연구의 한계를 극복하였다. 그러나 신규 불량 패턴을 탐지하기 위해서는 사전 불량 패턴에 대한 라벨링이 진행되어야 하며 In-distribution의 분포에 따라 신규 패턴 탐지 결과가 달라진다는 한계가 있다.

이같이 지도학습의 경우 데이터에 정확한 라벨링을 해야 한다는 단점이 있으며, 사전에 정의된 패턴 유형으로만 분류가 되어 새로운 패턴은 탐지하지 못하는 한계가 있다. 이에 본 연구에서는 데이터의 라벨을 필요로 하지 않는 비지도학습기반의 클러스터링 알고리즘을 이용하여, 새로운 패턴의 불량 유형과 이상치에도 능동적으로 탐지할 수 있게 한다.

2.2 비지도학습을 통한 패턴 클러스터링

Kang and Kang(2021)은 삼 네트워크(Siamese Network)를 이용하여 삼중항 손실(Triplet Loss) 학습을 통하여 클러스터링을 진행하였다. 데이터 변형을 통해 WBM과 변형된 WBM의 유사도를 최대화하였다. 하지만 이는 군집의 개수를 미리 정해야 하는 한계가 있다. 또한 스크래치 패턴의 경우 다른 패턴과 구분되지 못하였고, 다른 유형들 또한 명확하게 구분되지 못하였다. 이에 본 연구에서는 군집의 개수를 자동적으로 정하고, 자가지도학습을 이용하여 군집을 명확히 구분할 수 있는 잠재 벡터를 학습한다. 이를 통해 기존 패턴 유형에 한정되지 않고 새로운 패턴과 이상치 데이터를 능동적으로 탐지할 수 있게 한다.

Kim and Kang(2021)는 중앙값 필터(Median Filter)를 이용하여 WBM의 노이즈를 제거하고, 합성곱 오토인코더(Convolutional Autoencoder, CAE)를 이용하여 특징 추출을 진행하였다. 또한 디리클레(Dirichlet) 프로세스 기반 클러스터링을 진행하여 클러스터 개수를 자동으로 추출하였으며, 자가지도학습을 이용하여 잠재 벡터를 최적화하였다. 이 방법은 다양한 매개 변수에도 강건한 성능을 보여 다른 클러스터링 알고리즘보다 우수한 성능을 보였다. 이를 통해 클러스터 개수를 미리 정해야 하는 기존 연구의 한계를 극복하였고, 새로운 패턴을 능동적으로 탐지할 수 있게 되었다. 그러나 불균형한 불량 패턴 분포를 고려하지 않았고, 특히 얇고 긴 스크래치 패턴의 특징을 잘 보존하지 못하는 중앙값 필터 방법으로는 스크래치 패턴을 검출하지 못하는 한계가 있다. 이에 본 연구에서는 데이터 불균형을 고려한 클러스터링 방법을 제안한다. 또한 스크래치 패턴의 검출을 위해 깊이 우선 탐색(DFS) 알고리즘을 기반으로 노이즈 제거를 진행한다. 제안 방법론에 대한 자세한 설명은 다음 장에서 다루도록 한다.

3. 제안 방법론

이러한 앞선 연구의 한계점들을 보완하기 위해 본 연구에서는 공간 필터링(Spatial Filtering, SF)과 DFS 알고리즘을 활용하여

WBM의 노이즈 제거 후 CAE를 이용하여 특징 추출하는 과정을 통해 WBM의 스크래치 패턴을 극대화한다. 또한 디리클레 과정 혼합 모델(Dirichlet Process Mixture Model, DPMM)을 사용한 클러스터링 과정에서 클러스터 가중치(Weighted Cross Entropy, WCE)를 자가지도학습에 적용하여 데이터 불균형을 극복하고자 한다. 자세한 학습 과정은 다음 절에서 차례로 설명한다.

3.1 공간 필터링 및 깊이 우선 탐색 알고리즘 기반 노이즈 제거

WBM은 불량 패턴과 패턴에 영향을 주지 않는 노이즈 칩으로 구성되어 있기 때문에, 불량 패턴을 극대화하기 위해 노이즈로 간주되는 칩들을 제거하는 과정이 필요하다. 이를 위해 노이즈 제거는 두 단계로 이루어진다. 먼저 SF 방법을 활용하여 불량 패턴을 극대화한 후 DFS 알고리즘을 기반으로 노이즈 제거를 진행한다. SF는 불량 패턴을 극대화하기 위해 <Figure 2>와 같이 기준 칩을 중심으로 King Move Neighbors(KMN) 규칙을 이용하여 가중치를 계산하고, 가중치가 임계값(threshold)보다 크면 기준 칩을 불량 칩으로 대체한다. KMN 규칙은 Rook Move Neighbors(RMN) 규칙과 달리, 기준 칩과 맞닿아 있는 상하좌우뿐만 아니라 대각 관계에 있는 칩도 이웃으로 간주한다(Jeong *et al.*, 2008). 이는 더 넓은 영역을 필터링하여 본 연구에서 의도하는 불량 패턴을 극대화할 수 있도록 한다.

예를 들어 <Figure 3>의 (a)와 같이 기준 칩을 h_i , 주변 칩을 x_i 라고 할 때, 식 (1)과 같이 가중치(w)를 계산한다. 이때 x_i 가 불량인 경우 1, 아닌 경우 0의 값을 갖는다. 즉 h_i 의 w 는 $\frac{5}{9}$ 가 되고, 이때 threshold가 $\frac{5}{9}$ 이하라고 한다면 h_i 는 불량 칩으로 대체되어 <Figure 3>의 (b)와 같이 바뀐다.

$$w = \frac{1}{N} \times (x_1 + \dots + x_n),$$

$$h_i = \begin{cases} 1, & \text{if } w \geq \text{threshold} \\ x_i, & \text{otherwise} \end{cases} \quad (1)$$

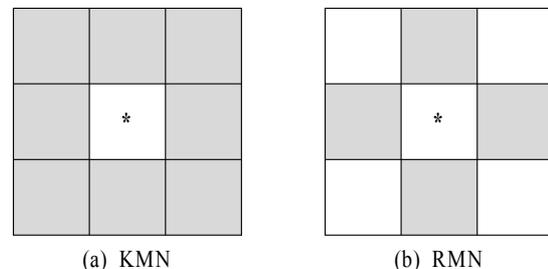


Figure 2. Neighborhood Construction Rules

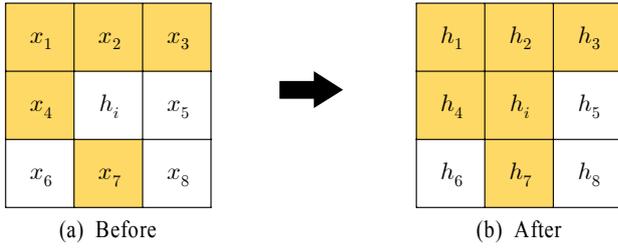


Figure 3. Result of Spatial Filtering

SF을 적용한 WBM을 DFS 알고리즘에 적용하기 앞서, <Figure 5>와같이 모든 불량 칩들에 대해 인덱스를 할당하였고, 총 22개의 불량 칩이 존재하는 것을 확인한다. <Figure 6>은 모든 불량 칩들의 연결 관계를 나타낸 인접 행렬이며, 식 (2)를 통해 계산된다. 여기서 $d(x_i, x_j)$ 는 i 번째 불량 칩과 j 번째 불량 칩 사이의 맨하탄 거리(Manhattan distance)를 의미한다.

$$A_{i,j} = \begin{cases} 1, & \text{if } d(x_i, x_j) \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

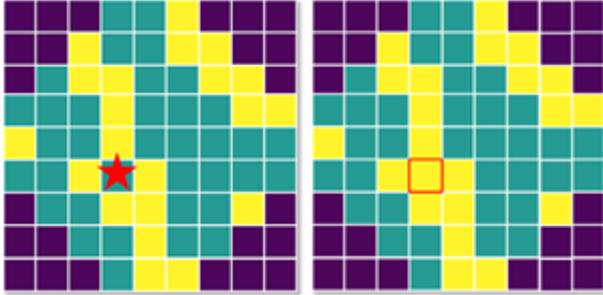


Figure 4. Result of WBM Spatial Filtering

이때 RMN 규칙을 이용하여 기준 칩을 중심으로 상하좌우에 위치한 불량 칩들만 이웃으로 간주한다. WBM 내 불량 칩들이 서로 이웃하면 인접행렬 $A_{i,j}$ 에 1의 값을 주었고, 그렇지 않으면 0의 값을 주었다. 이같은 방식으로 DFS 알고리즘을 적용하여 모든 칩들에 대해 불량 칩들과의 거리를 구할 수 있다.

본 연구에서는 이러한 SF 방법을 이용하여 불량 칩들 간 연결이 뚜렷하지 않은 패턴들의 특징을 극대화한다. 특히 스크래치 패턴 특성상 일부 패턴이 끊기는 경우가 다수 존재하기 때문에 이를 불량 칩으로 변경함으로써 <Figure 4>와 같이 스크래치 패턴이 극대화된다.

모든 칩들에 대한 거리 예시는 <Table 1>과 같다. 이때 거리란 단일 경로로 연결된 칩들 사이의 거리를 의미한다. 예를 들어 칩 1을 기준으로 칩 13까지의 거리는 10이고, 이는 칩 1과 칩 13은 불량 칩 10 개가 이어져 있음을 의미한다. 이같이 모든 불량 칩들에 대해 칩들간 거리를 계산할 수 있고, 계산된 모든 불량 칩들 간 거리는 임계값에 따라 <Figure 7>과같이 노이즈 제거 정도를 조절할 수 있다.

SF 방법을 이용해 불량 패턴을 극대화 한 후에는 DFS 알고리즘을 적용한다. DFS 알고리즘은 그래프 탐색 방법 중 하나로 기준 노드에서 다음 분기점으로 넘어가기 전에 해당 방향의 모든 루트를 완벽하게 탐색하는 방법이다(Tarjan, 1972). 즉, 시작 노드에서 진행할 수 있을 때까지 모든 하위 노드를 방문하며 더 이상 진행할 수 없다면 이전 분기점으로 돌아와 다시 다른 정점으로 탐색하는 방법이다. 이 방법은 노드와 연결되어있는 정점들을 깊게 탐색할 수 있다는 장점이 있다. DFS 알고리즘을 노이즈 제거 방식으로 사용함으로써 불량 패턴들이 얼마나 길게 연결되어 있는지 확인할 수 있다.

<Figure 7>의 (a)는 불량 칩 간의 연결이 6 미만인 모든 불량 칩들을 정상 칩으로 대체한 결과이고, (b)는 임계값을 7로 적용하여 노이즈를 제거한 결과를 보여준다. 임계값이 클수록 더 많은 불량 칩들이 정상 칩으로 대체되는 것을 확인할 수 있다. 이때 임계값을 결정하는 문제는 엔지니어의 지식에 의존한다(Huang et al., 2002). 또한 WBM내에 단일 불량 패턴이 존재한다고 가정했을 때에는 본 연구에서 설정한 임계값이 유용할 수 있으나, 실제 데이터에서는 하나 이상의 불량 패턴이 포함된 경우 즉, Mixed pattern을 위한 임계값의 조정이 필요하다.

<Figure 8>은 원본 WBM, 중앙값 필터를 이용했을 때의 결과, 그리고 SF+DFS 필터를 이용했을 때의 결과를 보여준다. 스크래치 패턴의 경우 중앙값 필터에 비해 SF+DFS 필터링을

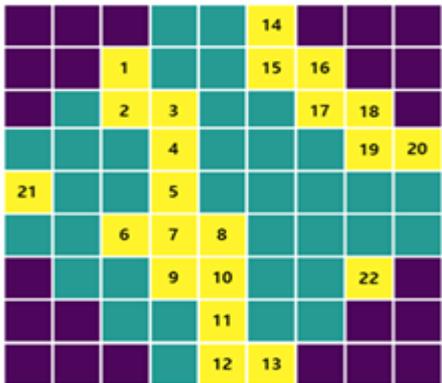


Figure 5. Example of Defect Chips Indexing

	1	2	3	4	5	6	22
1	0	1	0	0	0	0	0
2	1	0	1	0	0	0	0
3	0	1	0	1	0	0	0
4	0	0	1	0	1	0	0
5	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0

Figure 6. Example of Adjacency Matrix

Table 1. Example of Distance Matrix

Destination	Arrival				
	1	...	14
1	0	...	0
2	1	...	0
3	2	...	0
4	3	...	0
5	4	...	0
6	12	...	0
7	5	...	0
8	11	...	0
9	6	...	0
10	7	...	0
11	8	...	0
12	9	...	0
13	10	...	0
14	0	...	0
15	0	...	1
16	0	...	2
17	0	...	3
18	0	...	4
19	0	...	5
20	0	...	6
21	0	...	0
22	0	...	0
D	12	...	6

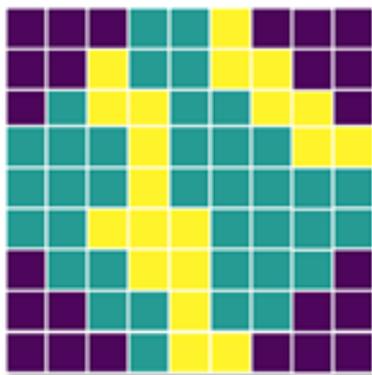
사용했을 때 특징이 극대화된 것을 확인할 수 있다. 또한 모든 라벨에서 불량 패턴에 영향을 주지 않는 노이즈 칩들이 제거되었으며, 불량 패턴 특징이 클러스터링에 용이하도록 추출된 것을 확인할 수 있다.

본 연구에서 제안한 SF+DFS 결과를 <Figure 9>와 같이 가시화할 수 있다. (a)는 필터링 전 원본 데이터이고, SF 방법을 이용해 (a)의 불량 패턴을 극대한 결과를 (b)를 통해 확인할 수 있다. 마지막으로 Algorithm 1의 DFS 필터링을 적용하여 불량 칩들 간의 연결이 12 미만인 노이즈 칩들을 제거한 결과 (c)와 같이 뚜렷해진 스크래치 패턴을 확인할 수 있다.

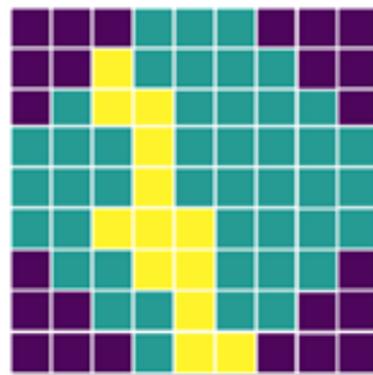
Algorithm 1. DFS Pseudo Code

```

1  Input : wafer bin map
2  for do
3      Adjacency network graph between defective chips of
4      need visited, visited = None
5      need visited.append(start node)
6      while need visited:
7          node = need visited.pop()
8          if node not in visited:
9              visited.append(node)
10             need visited.extend(graph[node])
11     if len(visited) < threshold:
12         = functional chips
13  Output : (denoised wafer bin map)
    
```



(a) Threshold : 6



(b) Threshold : 7

Figure 7. Example of DFS Filtering Results Based on Threshold

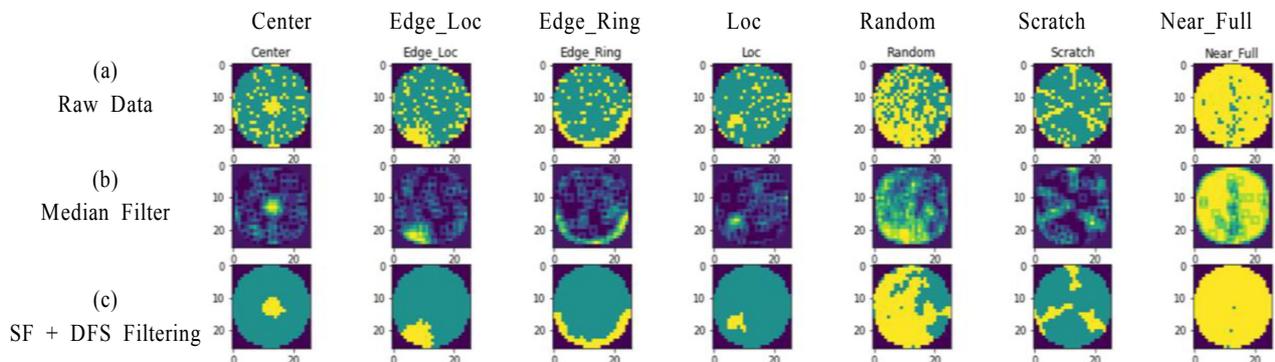


Figure 8. Result of Denoised WBM by Filtering Method

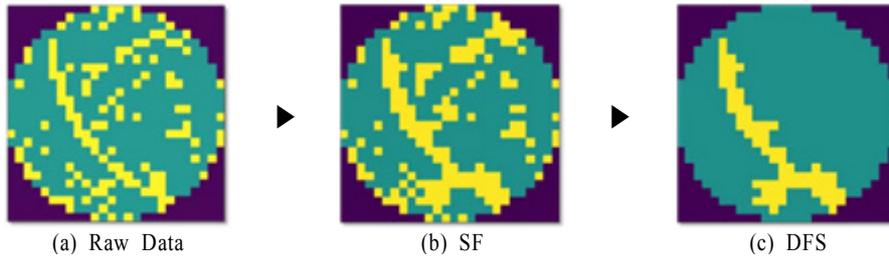


Figure 9. WBM Filtering Process

3.2 합성곱 오토인코더를 이용한 특징 추출

본 절에서는 SF+DFS 방법으로 노이즈가 제거된 WBM에 대하여 CAE를 이용해 각 불량 패턴을 구분하는 특징 추출 과정을 설명한다. CAE는 대표적인 신경망 기반 비지도학습 모델로, 주로 이미지 복원 분야에서 활용된다. 모델은 입력된 값을 그대로 예측해서 출력할 수 있도록 잠재 벡터(Latent Space)를 학습한다 (Guo et al., 2017). 오토인코더(Autoencoder, AE)는 Flatten 벡터를 사용하기 때문에 인코딩과 디코딩 과정에서 이미지의 공간적인 배열이나 이웃하는 픽셀들을 고려하지 못하는 한계가 있다. 반면에 CAE는 합성곱 필터를 사용하여 이웃하는 픽셀들 간의 관계를 유지함으로써 의미 있는 특징 추출이 가능하다. 즉, 시각적 및 공간적 구조를 보존하여 원본 이미지를 가장 잘 재구성하는 잠재 벡터를 찾기 위해 본 연구에서는 CAE를 이용하여 특징 추출을 진행한다(Kim and Kang, 2021).

CAE의 기본적인 동작 구조는 <Figure 10>과 같다. 모델의 인코더(Encoder)와 다운 샘플링(Down sampling)을 이용하여 데이터를 잠재 벡터에 압축하고, 이 잠재 벡터를 바탕으로 업 샘플링(Up sampling)을 진행하여 입력 값과 동일한 결과 값을 출력할 수 있게끔 디코더를 학습시킨다. 즉, 노이즈가 제거된 WBM을 CAE의 입력 값으로 사용하고, CAE의 인코더와 디코더를 바탕으로 WBM의 불량 패턴이 잘 표현되도록 잠재 벡터를 학습시킨다. 출력 값으로는 WBM의 특징이 반영된 잠재 벡터가 된다.

이때 손실 함수로는 식 (3)과 같은 범주형 교차 엔트로피 (Categorical Cross Entropy, CCE)를 사용하였다. CCE는 예측한 분포와 실제 분포의 차이를 바탕으로 손실을 추정하는 함수이며, 데이터를 실제 분포에 근접하도록 학습하는 장점이 있다.

$$Loss = - \sum_{i=1}^n y_i \cdot \log \hat{y}_i \quad (3)$$

여기서 y_i 는 실제 불량 패턴 레이블이며 \hat{y}_i 은 예측된 불량 패턴 레이블을 의미한다. 즉 실제 불량 패턴과 예측 불량 패턴을 최소화 하는 손실함수이다.

본 연구에서는 사전에 정의한 불량 패턴을 합성곱 신경망 모델에 학습한 후, 특징 추출기 부분을 이용해 각 불량 패턴의 특징을 추출한다. 각 WBM이 합성곱 신경망의 특징 추출기 부분을 통과하게 되면 각 WBM의 불량 패턴별로 구분되는 특징을 추출할 수 있다. 즉, 사전에 정의한 불량 패턴 샘플이 특징 추출기를 통과하면 사전에 정의한 불량 패턴끼리 서로 구분할 수 있는 특징이 추출되고, 새로운 불량 패턴 샘플이 특징 추출기를 통과하게 되면 기존 불량 패턴과 구분되는 특징을 추출할 수 있다.

3.3 불량 패턴 사이즈를 고려한 Fine Tuning

본 절에서는 CAE를 이용하여 추출된 잠재 벡터에 대해 디리클레 과정 혼합 모델(Dirichlet Process Mixture Model, DPMM)을 적용하여 클러스터링하는 방법을 설명한다. DPMM은 무한한 혼합 모델로, 각 불량 패턴이 가우시안 분포를 따른다고 가정한다. 이때 각 불량 패턴의 수는 데이터의 분포(우도)를 고려하여 사후 분포에 의해 확률적으로 결정된다. 즉, DPMM을 사용함으로써 각 불량 패턴의 수를 수동으로 결정해야하는 기존 연구의 한계를 극복할 수 있다. 이러한 DPMM의 유연성을 이용하여 본 연구에서는 WBM 마다 복잡한 불량 패턴을 포착하고, 새로운 불량 패턴 또한 동적으로 라벨링한다.

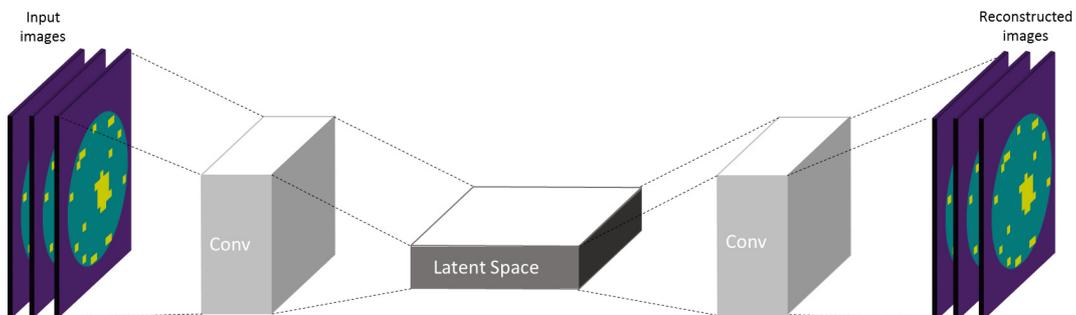


Figure 10. CAE Framework

본 연구에서는 Truncated Dirichlet Process(TDP)를 적용하여 학습 시간 단축과 학습의 효율성을 높였다. TDP는 각 클러스터의 분포가 독립적이며 가우시안 분포를 따르는 가정 하에, 데이터 내 분포 개수를 미리 가정하고 클러스터링을 진행하는 방법이다. 이를 이용해 데이터 개수가 증가할수록 TDP가 실제 디리클레 프로세스에 근사하도록 한다(Ishwaran and James, 2001). 본 연구에서는 잠재 벡터에 대해 디리클레 클러스터링을 진행하여 모든 WBM이 임시 클러스터 라벨을 할당한다(Kim and Kang, 2021). 이때 TDP 알고리즘에서는 매개변수 α 값을 조정하여 클러스터 생성의 민감도를 조절할 수 있다. 필요에 따라 작은 값으로 설정하여 소수의 불량 패턴을 감지할 수 있도록 한다. 본 연구에서는 α 값을 0.01로 설정하여 약 10%의 비율을 차지하는 소수의 스크래치 패턴을 효과적으로 클러스터링 한다. 즉, TDP 알고리즘의 매개변수 α 값을 조정하여 적은 불량 패턴이 검출될 수 있도록 한다.

또한 TDP를 이용하여 클러스터 레이블을 할당한 후에 자기지도학습 과정에서 불량 패턴 클러스터들의 사이즈 불균형 문제도 고려해야 한다. 라벨이 있는 경우에는 인위적으로 사이즈를 균형하게 맞출 수 있지만, 라벨이 없는 경우에는 이를 해결하기 어렵다. 만약 불량 패턴 클러스터들의 사이즈가 불균형하다면, 학습 시 비중이 높은 클러스터 위주로 학습이 진행되어 추후 모든 다수의 불량 패턴들이 소수의 클러스터로 편중될 가능성이 크다. 따라서 패턴의 비율을 고려하여 사이즈가 작은 클러스터들의 가중치를 높여 클러스터링을 진행해야 한다. 이를 통해 모든 불량 패턴들이 적절히 학습되고 추출될 수 있다(Longadge and Dongre, 2013).

본 연구에서는 이러한 불균형을 해소하기 위해 클러스터별로 가중치를 반영하여 식 (4)와 같은 가중치가 반영된 교차 엔

트로피(Weighted Cross Entropy, WCE)를 손실 함수로 사용한다. 이때 c 번째 클러스터의 가중치는 식 (5)와 같이 계산된다(Cui *et al.*, 2019).

$$WCE = - \sum_{i=1}^n \sum_{c=1}^C w_c \times I(y_i = c) \times \log \hat{y}_{i,c} \quad (4)$$

$$w_c = \frac{n}{C \times n_c} \quad (5)$$

여기서 w_c 란 각각의 Fine Tuning 단계에서 클러스터의 사이즈를 고려한 가중치이다. y_i 는 DPMM에서 결정된 불량 패턴 클러스터의 레이블이기 때문에, $I(y_i = c)$ 는 i 번째 웨이퍼가 클러스터 c 에 속하면 1 아니면 0이 되는 것을 의미하고 $\hat{y}_{i,c}$ 은 그 확률 값을 의미한다. n 은 전체 WBM 데이터 개수이며 C 는 디리클레 과정으로 인해 추정된 클러스터의 개수이며 n_c 는 c 클러스터에 있는 샘플의 개수이다. 즉, n_c 가 작을수록 w_c 가 증가하며 n_c 가 클수록 w_c 가 감소한다. 이를 통해 클러스터의 사이즈가 클러스터 가중치로 반영될 수 있다.

다시 말해서, 데이터 개수가 많은 클러스터에는 작은 가중치를 두고, 데이터가 적은 클러스터에는 큰 가중치를 적용하는 페널티 항을 둔다. 이를 통해 많은 데이터로 이루어져 있는 클러스터의 학습 참여율을 제한하고 적은 데이터로 이루어져 있는 클러스터의 학습 참여율을 증가시킴으로써 클러스터 비율에 상관없이 모든 클러스터가 고르게 학습되도록 한다.

TDP를 통해 임시 클러스터링 라벨을 예측하게끔 잠재 벡터를 재학습을 시켰으며, 이와 동시에 미세 조정 과정에서 WCE를 반영하여 <Figure 11>과같이 클러스터 간 경계가 뚜렷해지며 정확한 클러스터링을 진행할 수 있도록 한다.

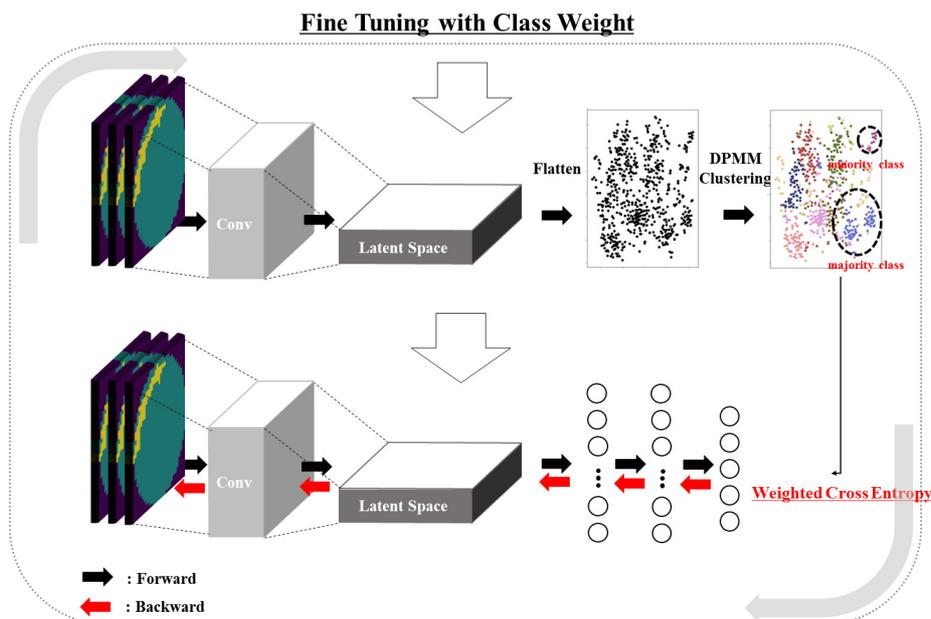


Figure 11. Fine Tuning Process Framework Using Cluster Weight

4. 실험 및 평가

4.1 데이터 소개

본 연구에서는 캐글(Kaggle)의 WM-811K 데이터 셋을 사용한다. 총 811,457개의 WBM 데이터가 존재하며 이 중 라벨이 있는 데이터는 170,950개이다. 불량 유형으로는 Center, Donut, Edge Loc, Edge Ring, Loc, Random, Scratch, Near full, None의 9가지로 분류된다. 실제 안정화된 공정에서는 불량 WBM에 비해 정상 WBM이 절대적으로 많이 나타날 수 있지만, 본 연구에서는 불균형한 불량 패턴 분포를 고려하지 않고, 상대적으로 적은 클래스의 패턴을 검출하지 못하는 기존 연구의 한계점(Kim and Kang, 2021)을 보완하여 소수의 불량 패턴 극대화라는 제안 방법론의 우수성을 보이기 위해, 정상 웨이퍼로 분류되는 None 패턴의 데이터는 사용하지 않았으며, 26×26 크기의 WBM만 사용한다. 또한 Donut 유형의 경우 데이터가 1개만 존재하여 제거한다. 이에 따라 본 실험에 사용된 데이터는 총 875개이며 7개의 불량 패턴이 존재한다. <Figure 12>에서 볼 수 있듯이 Loc, Edge Loc 라벨의 비율이 높으며 Scratch, Edge Ring, Near Full 라벨의 비율이 낮은 불균형한 데이터 셋이다. 모델을 정량적으로 평가하기 위해 875개의 데이터를 80%의 훈련 데이터와 20%의 검증 데이터로 분류하였다.

4.2 노이즈 제거 및 특징 추출

본 실험에서 노이즈 제거 시 사용한 SF와 DFS 알고리즘의 임계값은 <Table 2>와 같다. 먼저 SF 필터링을 진행할 때 기준 칩 주변으로 4개 이상의 불량 칩이 존재하면 정상 칩을 불량 칩으로 대체한다. 또한 DFS 필터링의 임계값은 12로 하였다

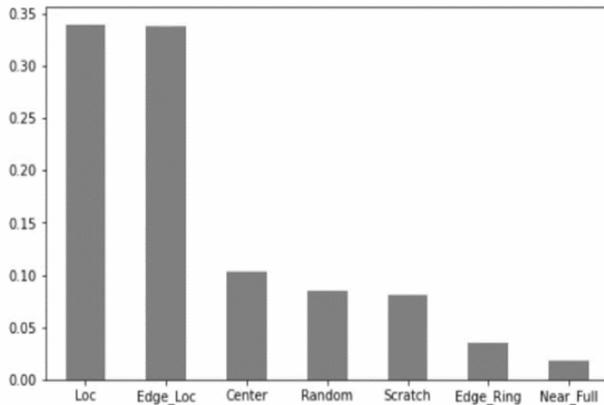


Figure 12. Proportion of WBM Defect Patterns

Table 2. Hyperparameter by Filtering Method

Process	Hyperparameter	Value
SF Filtering	threshold	4
DFS Filtering	threshold	12

Table 3. CAE Structure

Layer	Filter Size	Output Size
X	-	64×64×3
Conv1	128×3×3	64×64×128
MaxPool	-	32×32×128
Conv2	64×3×3	32×32×64
MaxPool	-	16×16×64
Conv3	32×3×3	16×16×32
MaxPool	-	8×8×32
Conv1 ^t	32×3×3	8×8×32
UpSample	-	16×16×32
Conv2 ^t	64×3×3	16×16×64
UpSample	-	32×32×64
Conv3 ^t	128×3×3	32×32×128
UpSample	-	64×64×128
X ^t	3×3×3	64×64×3

(Kim et al, 2018). Kim et al(2018)은 반도체 공정 전문가 의견에 따라 DFS 필터링 시 12가 최적의 임계값으로 설정하였다. 즉 불량 칩들이 12개 미만으로 이어져 있을 경우 이어져 있는 모든 칩들을 정상 칩으로 대체한다.

또한 특징 추출을 위해 사용된 CAE 모델의 구조는 <Table 3>과 같다. 효율적인 딥러닝 학습을 위해 26×26의 WBM 크기를 64×64로 조정하였다. 인코더와 디코더 각각 3개의 합성곱층을 사용하였으며 인코더의 각 층별로 Max Pooling을 진행하였으며 디코더는 각 층마다 업 샘플링을 진행한다. 최적화 방법으로는 Adam Optimizer를 사용한다(Kingma and Ba, 2014).

4.3 클러스터링 및 미세 조정

일반적으로 사전지식 없이 클러스터링을 수행할 때는 다양한 지표(metric)를 기반으로 가장 적합한 클러스터 개수를 결정한다. 그러나 본 연구에서는 기존 연구의 한계를 보완하여 소수의 불량 패턴을 극대화하는 새로운 방법론을 제안한다는 점을 강조하기 위해 <Table 4>와 같이 기존 연구와 동일한 실험 조건에서 실험을 수행한다. 실험에서는 총 5회 미세 조정을 진행하며, 데이터 내에 10개의 분포가 있다고 가정한다. 또한 완전 공분산 형태를 가정한다. α 값은 클러스터의 정도를 결정하는 파라미터이다. α 값이 클수록 적은 데이터로 구성된 많은 클러스터가 생성된다. 본 실험에서는 α 를 0.01로 사용한다(Kim and Kang, 2021).

Table 4. DPMM Clustering Hyperparameter

Process	Hyperparameter	Value
Fine Tuning	n	5
TDP	N_components	10
	Covariance_type	Full
	α	0.01

4.4 클러스터링 결과

클러스터링 결과 총 10개의 클러스터가 형성되었으며 <Figure 13>과 <Figure 14>는 종합적인 클러스터링 결과를 보여준다. <Figure 13>는 가장자리에 불량 발생 WBM들이 클러스터링 된 것을 확인할 수 있다. 이는 박막 증착 공정 과정에서 불량 발생했을 가능성이 높음을 시사한다. 또한 <Figure 14>는 중앙에 큰 불량 발생 WBM들이 클러스터링 되었다. 이는 식각 공정에서 불량 발생했을 가능성이 높다.

4.5 평가 방법 및 제안 방법론과의 비교

(1) 평가 방법

본 실험에서는 클러스터링의 정확도를 정량적으로 평가하

기 위해 잠재 벡터를 평가하는 방법을 사용한다(Caron et al, 2018). 최적화된 잠재 벡터는 클러스터 간의 경계가 뚜렷하게 구분되는 고품질 벡터라고 가정하며, 클러스터링과 미세 조정을 통해 이러한 잠재 벡터를 얻을 수 있다. 이에 따라 학습된 잠재 벡터의 분류 성능을 평가 지표 사용한다. 즉, 분류 성능이 우수한 잠재 벡터일수록 클러스터링 결과가 더욱 정확하게 나타나는 것으로 가정한다.

먼저 자가지도학습을 이용한 미세 조정 단계에서는 잠재 벡터를 최적화한다. 역전파 알고리즘을 이용하여 학습하며, 데이터의 입력부터 잠재 벡터까지의 층은 학습이 되지 않게 고정하여 잠재 벡터의 업데이트를 방지한다. 이후 <Figure 15>와 같이 분류 레이어만 추가하여 분류를 진행한다(Kim and Knag, 2021). 분류 성능 지표로는 정밀도(Precision), macro avg,

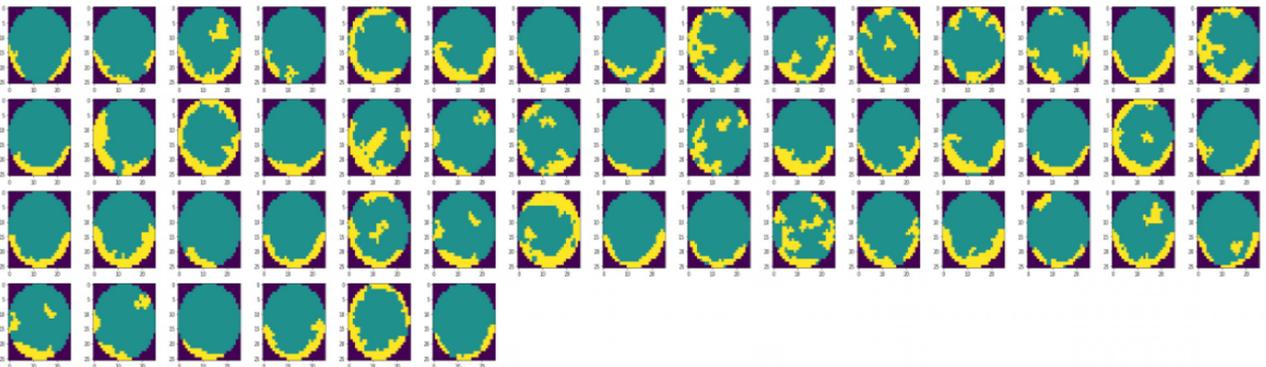


Figure 13. Examples of Edge Ring Defect Pattern Cluster

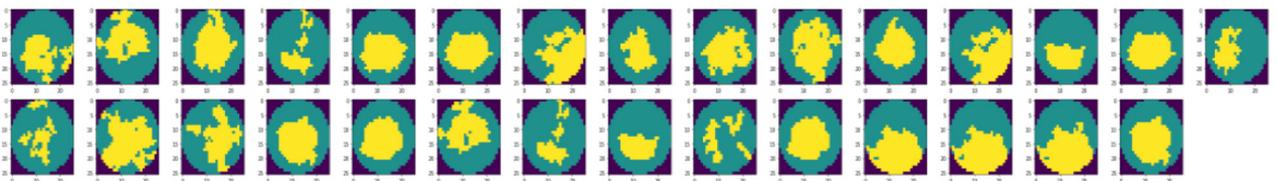


Figure 14. Examples of Center Defect Pattern Cluster

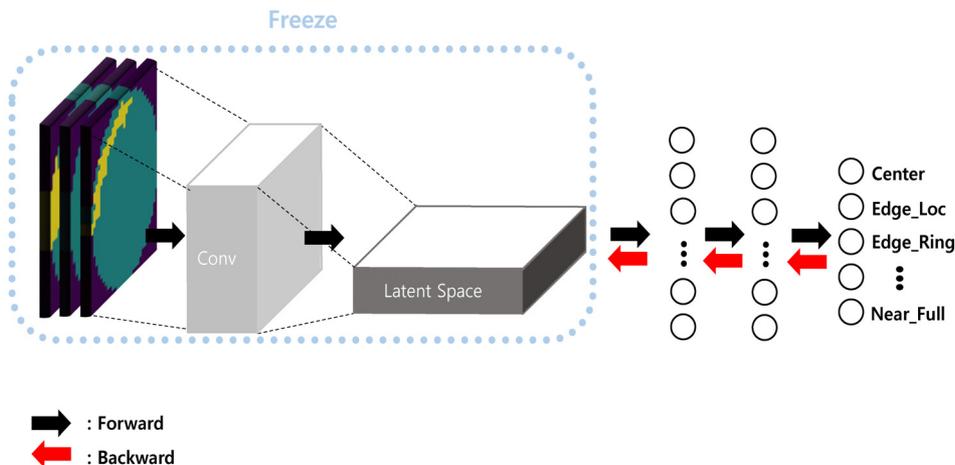


Figure 15. Clustering Evaluation Process Framework

weighted macro avg를 사용하였으며 식 (6)~식 (8)과 같이 계산된다. 제안 방법론을 통해 생성된 각 클러스터는 실제 레이블을 의미하지 않는다. 즉, 실제 레이블과 같지 않은 상황과 정상 WBM들을 배제한 조건에서 precision을 평가지표로 사용하는 것이 타당하지 않을 수 있으나 기존 연구와 비교하여 제안 방법론의 우수성을 부각하기 위해 동일한 평가지표를 사용한다.

$$\alpha_i = \frac{TP_i}{TP_i + FP_i} \quad (6)$$

$$Macro\ Average\ Precision = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + FP_i} \quad (7)$$

$$Weighted\ Average = \frac{\sum_{i=1}^n \alpha_i n_i}{\sum_{i=1}^n n_i} \quad (8)$$

여기서, TP_i : True Positive, FP_i : False Positive.

이때 α_i 는 i 번째 클래스의 정밀도, n_i 는 클래스 i 의 데이터 수를 나타낸다. 그리고 *Macro Average Precision*은 각 클래스의 정밀도의 평균, *Weighted Average*는 클래스 내 데이터 개수를 고려한 정밀도의 가중 평균을 의미한다(Opitz and Burst, 2019).

(2) 제안 방법론과의 성능 비교

<Table 5>는 첫 번째 미세 조정 과정에서 형성된 클러스터와 각 클러스터 가중치를 보여준다. 첫 번째 미세 조정에서 10개의 클러스터가 형성되었으며, 각 클러스터 내 데이터 개수와 데이터 비율은 N , P 와 같다. 클러스터 별 가중치(*Weight*)를 보면 데이터 비율인 P 가 높을수록 가중치가 작은 것을 확인할 수 있다. 예를 들어 데이터 비율이 0.04로 가장 작은 클러스터 9의 가중치를 보면 2.41로 가장 큰 가중치를 가지는 것을 확인할 수 있다. 반대로 비율이 0.19로 가장 큰 클러스터 3의 가중치는 0.54로 가장 작은 가중치를 가진다. 이와 같이 미세 조정마다 클러스터 가중치를 조정함으로써 데이터 불균형에 강건하게 학습을 진행한다.

Table 5. Cluster Weight in First Fine Tuning

Cluster	N	P	Weight
1	43	0.06	1.63
2	63	0.09	1.11
3	130	0.19	0.54
4	135	0.19	0.52
5	30	0.04	2.33
6	49	0.07	1.43
7	48	0.07	1.46
8	75	0.11	0.93
9	29	0.04	2.41
10	98	0.14	0.71

<Figure 16>~<Figure 18>과 <Table 6>~<Table 8>은 각각 필터링 방법에 따른 성능 차이, 클러스터 가중치 적용 시 성능 차이, 기존 연구 방법론과의 성능 차이를 가시화한다. 또한 실험을 동일한 조건으로 5회 반복하여 표준 편차를 계산한다. 작은 표준 편차는 모델의 안정성이 높아 강건한 모델을 의미하며, 큰 표준 편차는 모델의 불안정성이 높아 불안정한 모델을 의미한다.

<Figure 16>과 <Table 6>은 보편 중앙값 필터를 적용했을 경우와 SF+DFS 필터링을 적용했을 때의 성능 차이를 보여준다. SF와 DFS 알고리즘 적용 결과 모든 라벨에 대해 정밀도가 향상되었다. 특히 Center 패턴의 경우 0.490에서 0.746으로 약 52.2%가 향상되었으며 Scratch 패턴의 경우에도 0.464에서 0.538로 약 15.9%가 향상되었다. 하지만 고리형 패턴의 특성상 DFS 알고리즘 적용 시 임계값에 미치지 못하는 경우 불량 칩이 정상 칩으로 대체되어 검출율이 다소 하락하는 경우가 발생하였으며 이로 인해 표준편차가 다소 증가하였다. 그러나 5번의 미세조정 후 전체적인 정밀도는 중앙값 필터를 적용했을 경우보다 향상되었다. Random 불량 패턴의 경우도 마찬가지로 정상 패턴을 불량 패턴으로 바꾸는 SF의 특징 상 Random 불량 유형이 None에 가까운 불량 패턴을 형성하게 되는 경우가 발생하였다. 이로 인해 검출율이 다소 떨어지는 경우가 발생하였으며 Random 패턴의 정밀도에 대한 표준편차가 증가하였다. 하지만 전체적인 정밀도는 중앙값 필터의 경우보다 향상되었다. Weighted average 또한 중앙값 필터보다 DFS 알고리즘 기반 필터의 경우가 정밀도가 0.576에서 0.690으로 약 19.8% 증가하였으며 더 안정적인 것을 확인할 수 있다.

Table 6. Precision Comparison by Filtering Method

Filter	Median	SF + DFS
Center	0.490 (0.191)	0.746 (0.084)
Edge-Loc	0.616 (0.068)	0.738 (0.013)
Edge-Ring	0.520 (0.114)	0.546 (0.264)
Loc	0.520 (0.083)	0.658 (0.049)
Random	0.464 (0.135)	0.678 (0.211)
Scratch	0.464 (0.125)	0.538 (0.066)
Near-Full	0.830 (0.087)	1.000 (0.000)
Macro Average	0.678 (0.396)	0.702 (0.029)
Weighted Average	0.576 (0.073)	0.690 (0.036)

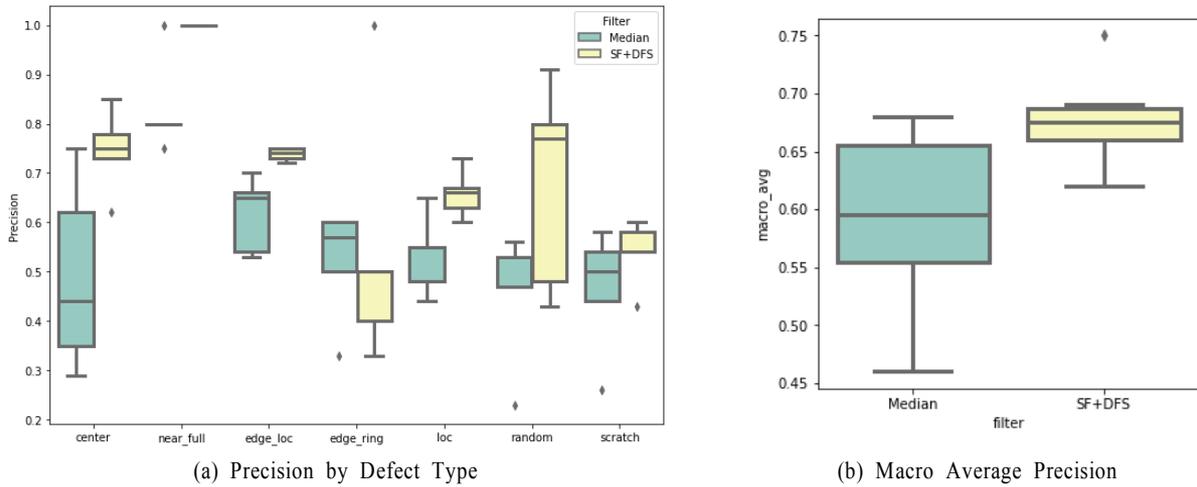


Figure 16. The Precision Box Plot by Filtering Method

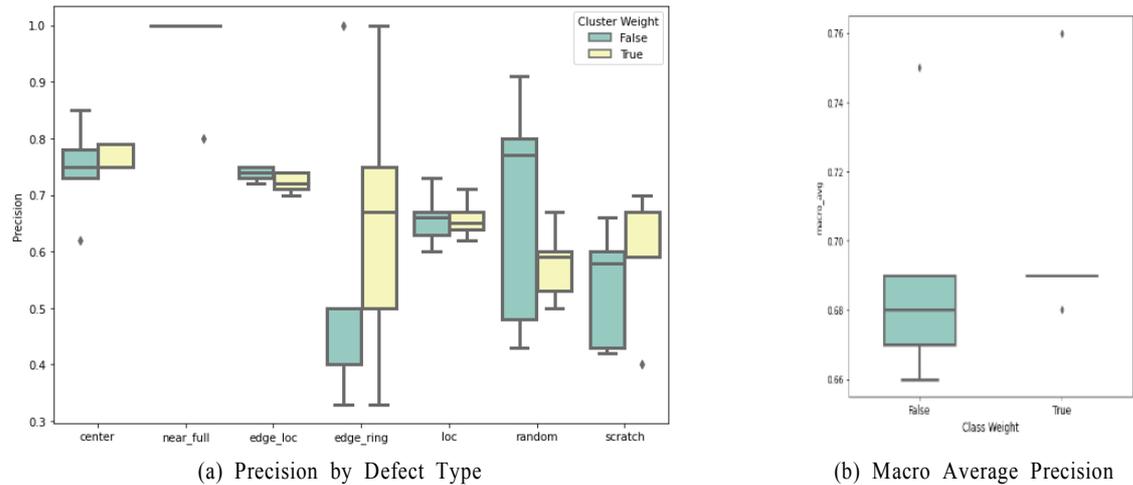


Figure 17. The Precision Box Plot by Using Clustering Weight

또한 표준편차를 보면 Edge-Ring 및 Random 패턴의 경우를 제외하고 모든 패턴에서 표준편차의 평균이 감소한 것을 볼 수 있다. Weighted Average의 표준편차 평균의 경우 0.073에서 0.036으로 약 58%가 감소한 것을 알 수 있다. 이는 중앙값 필터에 비해 공간 필터링 및 DFS 알고리즘을 노이즈 제거 방식에 적용할 경우 불량 패턴 별 검출 결과의 편차가 크지 않고 강건한 것을 의미한다. 하지만 Edge-Ring의 경우 DFS 알고리즘 적용 결과 불량 패턴이 끊기는 경우가 존재하여 테스트 데이터셋에 따라 불량 패턴 검출율이 하락한 경우가 발생하였다. Random 패턴도 마찬가지로 SF 및 DFS 알고리즘 적용 결과 Near-Full에 가까운 패턴을 형성하는 경우가 발생하여 정밀도가 하락한 경우가 존재하였으며 이에 따라 표준편차가 증가하였다. 하지만 전체적인 표준편차의 감소율을 고려하면 중앙값 필터에 비해 SF 및 DFS 알고리즘을 노이즈 제거에 적용하였을 때 더 안정적임을 알 수 있다. 따라서 추후 실험에서는 공간 필터링 및 DFS 알고리즘 기반의 필터링 방법을 적용하였다.

Table 7. Precision Comparison by Cluster Weight

Cluster Weight	False	True
Center	0.746 (0.075)	0.766 (0.020)
Edge-Loc	0.738 (0.012)	0.722 (0.016)
Edge-Ring	0.558 (0.236)	0.650 (0.227)
Loc	0.640 (0.044)	0.658 (0.031)
Random	0.728 (0.189)	0.578 (0.060)
Scratch	0.438 (0.088)	0.578 (0.104)
Near-Full	1.000 (0.000)	0.960 (0.080)
Macro Average	0.690 (0.033)	0.702 (0.029)
Weighted Average	0.680 (0.019)	0.690 (0.006)

<Figure 17>과 <Table 7>은 클러스터 가중치 반영 전후의 성능 차이를 나타낸다. Random의 경우를 제외하고 모든 패턴에 대해서 정확도가 비슷하거나 상승한 것을 볼 수 있다. 특히 클래스의 비율이 낮은 Edge Ring의 경우 정밀도가 16.5% 상승했으며 스크래치의 경우 32% 상승하여 모두 비약적으로 상승한 것을 확인할 수 있다. 클러스터 가중치를 반영함으로써 데이터 불균형을 보정하여 정밀도가 상승한 것을 확인할 수 있다.

또한 표준편차를 보면 Center, Edge-Ring, Loc, Random, Near-Full 패턴의 경우 클러스터 가중치 적용 결과 표준편차의 평균이 감소한 것을 알 수 있다. 이 외 Edge-Loc, Scratch, Near-Full의 경우 미세하게 상승하였지만, Weighted Average의 표준편차 평균이 0.019에서 0.006로 약 68% 감소한 것을 알 수 있다. 이를 통해 클러스터 가중치 적용 결과 모델이 안정적이고 편차가 작아진 것을 알 수 있다. 즉 패턴 불균형에 강건하고 비율이 적은 불량 패턴도 안정적이고 효율적으로 검출할 수 있음을 나타낸다.

Table 8. MSE Comparison by Cluster Weight

Value	Median		Average	
	False	True	False	True
Cluster Weight				
Center	0.750	0.750	0.746	0.766
Edge-Loc	0.750	0.720	0.738	0.722
Edge-Ring	0.500	0.670	0.558	0.650
Loc	0.660	0.650	0.640	0.658
Random	0.770	0.590	0.728	0.578
Scratch	0.580	0.590	0.438	0.578
Near-Full	1.000	1.000	1.000	0.960
Mean	0.714	0.710	0.693	0.702
MSE	0.023	0.015	0.027	0.014
Decrease Rate	34.35%		48.15%	

<Table 8>은 클러스터 가중치 적용 여부에 따른 5번의 반복 실험 결과 얻어진 각 라벨 별 Median, Average 값의 평균 제곱 오차(Mean Squared Error, MSE)로 식 (8)과같이 계산한다.

$$MSE = \sum_{i=1}^n (y_i - \bar{y})^2 \tag{8}$$

여기서 y_i 는 반복 실험 결과 얻어진 각 클래스 별 Median Precision 및 Average Precision이다. \bar{y} 는 7개의 모든 클래스에 대한 정밀도 평균이다. 따라서 MSE 값이 클수록 클래스 간의 예측 편차가 크을 의미하며 작을수록 클래스 간의 예측이 고르게 이루어진 것을 의미한다. <Table 8>을 살펴보면 클러스터 가중치를 적용했을 경우 Median 값과 Average 값 모두 MSE가 각각 34.35%, 48.15% 감소한 것을 알 수 있다. 클러스터 가중치 적용 결과 클러스터의 비중에 관계없이 고르게 예측했음을 의미한다. 즉 비중이 높은 클러스터의 예측율과 비중이 낮은 클러스터 예측율의 차이가 감소하였으며 클러스터 비율에 강건한 모델임을 알 수 있다. 그러나 소수 클러스터인 Random의 경우 실제 성능(Precision)은 감소하였고, 다수 클러스터인 Edge-Loc과 Edge-Ring의 성능이 증가한 것도 확인된다. 이는 기존 방법론과 동일한 조건의 filtering 임계값을 사용한 한계로, 실제 레이블을 의미하지 않을 수 있다.

또한 <Figure 18>과 <Table 9>는 기존 연구방법론과 제안 방법론의 성능 차이를 나타낸다. 중앙값 필터와 가중치를 반영하지 않은 클러스터를 사용한 기존 연구보다 SF+DFS 알고리즘 기반 필터링과 가중치를 반영한 클러스터링을 상용한 제안 방법의 경우 모든 라벨에서 정밀도가 상승한 것을 확인할 수 있다. 또한 Weighted Average의 경우 0.588에서 0.690으로 17.3% 상승하였다. 표준편차의 경우 Edge-Ring 및 Near-Full 패턴의 경우를 제외하고 모든 패턴에 대해서 표준편차의 평균이 감소한 것을 확인할 수 있다. 하지만 Edge-Ring 및 Near-Full 패턴의 경우 테스트 데이터 셋이 적어 1개의 오 분류에도 영향

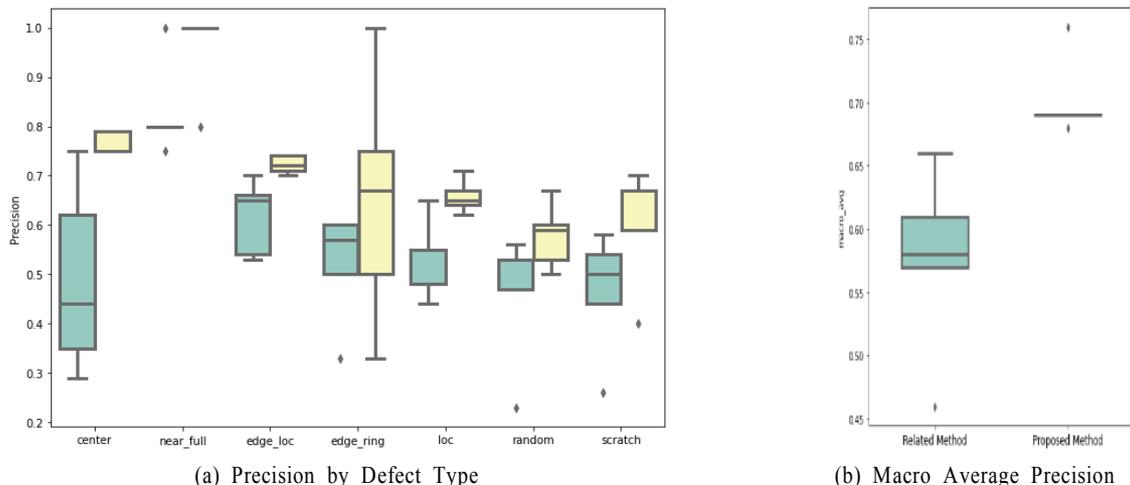


Figure 18. The Precision Box Plot Related Method and Proposed Method

Table 9. Comparison of Precision between Median Filter and SF+DFS Filter

Method	Filter	Cluster Weight	Filter	Cluster Weight
	Median	False	SF+DFS	True
Center	0.490 (0.171)		0.766 (0.020)	
Edge-Loc	0.608 (0.068)		0.722 (0.016)	
Edge-Ring	0.525 (0.102)		0.650 (0.227)	
Loc	0.530 (0.074)		0.658 (0.031)	
Random	0.523 (0.121)		0.578 (0.060)	
Scratch	0.515 (0.112)		0.578 (0.104)	
Near-Full	0.850 (0.009)		0.960 (0.080)	
Macro Average	0.576 (0.066)		0.702 (0.029)	
Weighted Average	0.588 (0.067)		0.690 (0.006)	

을 많이 받는 문제점이 존재한다. 또한 앞서 말했듯이 Edge-Ring의 경우 DFS 알고리즘에 따라 패턴이 끊겨 검출력이 떨어지는 경우가 발생하였다. 하지만 전체적인 Weighted Average 정밀도의 표준편차의 평균을 보면 0.067에서 0.006으로 약 91%가 감소하였다. 이는 기존 연구에서 중앙값 필터와 클러스터 가중치를 적용하지 않았을 때 보다 SF 및 DFS 알고리즘으로 노이즈 제거를 하며 클러스터 가중치를 적용했을 경우 모델이 더욱 강건하다는 것을 의미한다.

따라서 본 연구에서 제안하는 SF+DFS 알고리즘을 WBM 노이즈 제거에 적용한 결과 불량 패턴이 뚜렷해지며 검출에 용이해짐을 확인할 수 있다. 또한 클러스터 가중치 적용 결과 클러스터 비율에 상관없이 클러스터 간 예측 편차가 감소하며 모델이 강건해짐을 확인한다.

5. 결론

반도체 EDS 테스트를 통해서 웨이퍼 내의 양품 여부와 위치 정보를 나타내는 WBM을 추출할 수 있다. WBM은 다양한 불량 패턴을 가지며 같은 불량 패턴을 가지는 WBM은 같은 공정에서 불량이 가능했을 가능성이 높다. 따라서 불량 원인을 사전에 탐지하고 해결하여 생산성 및 효율성 향상을 통해 수율 개선 효과를 기대할 수 있다.

기존 연구의 대부분은 WBM 불량 패턴의 라벨을 활용하는 지도학습이다. 하지만 모든 데이터에 정확하고 고품질의 라벨링을 하는 것은 시간과 비용이 매우 많이 소모되는 작업이다. 또한 지도학습 특성 상 기존에 정의된 불량 유형으로만 분류한다는 한계가 있다. 따라서 새로운 불량 패턴의 탐지가 어렵다는 한계가 있다. 이후 진행된 불량 패턴 클러스터링 연구에서는 자가지도학습을 이용하여 불량 패턴의 개수를 자동적으로 추출하는 연구가 진행되었다. 하지만 스크래치 패턴을 거의 검출하지 못하였으며, 불량 패턴 데이터 사이즈의 불균형을 고려하지 못한다는 한계를 지니고 있다. 즉 전체 데이터 중에서 차지하는 비율이 높은 클래스 위주로 학습이 진행되어 비율이 낮은 클래스는 학습에 영향을 끼치지 못하였다.

이를 해결하기 위해서 본 연구에서는 클러스터 가중치를 이용하여 페널티 항을 주었다. 즉 비율이 높은 클래스에는 적은 가중치를 주어 학습 참여율을 규제하고 비율이 낮은 클래스에는 높은 가중치를 주어 학습에 적극적으로 참여할 수 있도록 하였다. 즉, 학습 시에 Weighted Cross Entropy를 손실 함수로 사용하여 데이터 불균형에 강건한 모델을 구현한다. 실험 결과 상대적으로 데이터의 개수가 적은 Edge-Ring, 스크래치 패턴의 검출율이 향상되었다. 제안 방법을 통해 클러스터링 알고리즘을 이용하여 스크래치 패턴을 효과적으로 검출할 수 있게 되었으며, 동시에 데이터 불균형을 고려한다.

또한 본 연구에서는 패턴을 극대화하기 위하여 DFS 알고리즘을 기반 필터링 방식을 이용한다. 이를 통해 얇고 긴 스크래치 패턴을 효과적으로 검출할 수 있게 되었고, 중앙값 필터와 클러스터 가중치를 적용하지 않은 기존 연구에 비해 모든 라벨에서 정확도가 상승하였다.

하지만 본 연구에는 다음과 같은 한계가 존재한다. 먼저 필터링 과정에서 정상 칩을 불량 칩으로 대체하는 과정에서 Random 클래스가 패턴을 형성하게 되는 경우가 존재한다. 이로 인해 Random 클래스가 Near-Full과 유사하게 필터링 되는 경우가 존재한다. 따라서 추후 연구에서는 사전에 Random 패턴을 규정하고 Random 패턴의 경우 선택적으로 필터링을 방법을 강구해야 한다.

참고문헌

- Alawieh, M., Boning, D., and Pan, D. (2020), Wafer Map Defect Patterns Classification using Deep Selective Learning, *ACM/IEEE Design Automation Conference (DAC)*, 1-6.
- Baek, S. H., Lee, C. H., and Kim, S. B. (2020), Detection of a New Pattern of Wafer Bin Maps Using Deep Learning, *Korean Journal of Industrial Engineering*, **46**(3), 326-337.
- Cui, Y., Jia, M., Lin, T. Y., Song, Y., and Belongie, S. (2019), Class Balanced Loss Based on Effective Number of Samples, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9268-9277.
- Guo, X., Liu, X., Zhu, E., and Yin, J. (2017), Deep Clustering with

- Convolutional Autoencoders, *International Conference on Neural Information Processing*, 373-382.
- Hansen, M., Nair, V., and Friedman, D. (1997), Monitoring Wafer Map Data from Integrated Circuit Fabrication Processes for Spatially Clustered Defects, *Technometrics*, **39**(3), 241-253.
- Huang, C. J., Wu, C. F., and Wang, C. C. (2002), Image Processing Techniques for Wafer Defect Cluster Identification, *IEEE Design & Test of Computers*, **19**(2), 44-48.
- Ishwaran, H. and James, L. (2001), Gibbs Sampling Methods for Stick-Breaking Priors, *American Statistical Association*, **96**(453), 161-173.
- Jeong, Y., Kim, S., and Jeong, M. (2008), Automatic Identification of Defect Patterns in Semiconductor Wafer Maps Using Spatial Correlogram and Dynamic Time Warping, *IEEE Transactions on Semiconductor Manufacturing*, **21**(4), 625-637.
- Kahng, H. and Kim, S. (2021), Self-Supervised Representation Learning for Wafer Bin Map Defect Pattern Classification, *IEEE Transactions on Semiconductor Manufacturing*, **34**(1), 74-86.
- Kang, H. and Kang, S. (2021), Unsupervised Representation Learning based on Triplet Loss for Wafer Map Pattern Analysis, *Korean Institute of Information Scientists and Engineers*, **48**(1), 555-5511.
- Kang, S. (2020), Rotation Invariant Wafer Map Pattern Classification With Convolutional Neural Networks, *IEEE Access*, **8**, 170650-170658.
- Kim, B., Jeong, Y., Tong, S., Chang, I and Jeong, M. (2015), A Regularized Singular Value Decomposition-Based Approach for Failure Pattern Classification on Fail Bit Map in a DRAM Wafer, *IEEE Transactions on Semiconductor Manufacturing*, **28**(1), 41-49.
- Kim, D. and Kang, P. (2021), Dynamic Clustering for Wafer Map Patterns Using Self Supervised Learning on Convolutional Autoencoders, *IEEE Transactions on Semiconductor Manufacturing*, **39**(3), 444-454.
- Kim, J., Lee, Y., and Kim, H. (2018), Detection and Clustering of Mixed-type Defect Patterns in Wafer Bin Maps, *Iise Transactions*, **50**(2), 99-111.
- Ko, H., Park, S., Ryu, S., Kim, S., Lee, G., Lee, D., Pae, S., Lee, E., Ji, Y., Jinag, H., Jeong, T., Uemura, T., Kwon, D., Do, H., Kahng, H., Cho, Y., Lee, J., and Kim, S. (2020), Early Diagnosis and Prediction of Wafer Quality Using Machine Learning on sub-10nm Logic Technology, *IEEE International Reliability Physics Symposium (IRPS)*, 1-5.
- Liu, S. F., Chen, F. L., and Lu, W. B. (2002), Wafer Bin Map Recognition Using a Neural Network Approach, *International Journal of Production Research*, **40**(10), 2207-2223.
- Longadge, R. and Dongre, S. (2013), Class Imbalance Problem in Data Mining Review, arXiv preprint arXiv:1305.1707.
- Nakazawa, T. and Kulkarni, D. (2018), Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network, *IEEE Transactions on Semiconductor Manufacturing*, **31**(2), 309-314.
- Optiz, J. and Burst, S. (2019), Macro F1 and Macro F1, arXiv preprint arXiv:1911.0334.
- Park, S., Park, H., Kim, J., Lee, C., Kim, Y., and Kim, S. (2018), Deep Learning-based Predictive Maintenance for Smart Manufacturing, *Journal of Manufacturing Systems*, **48**, 144-156.
- Song, C., Jeong, Y., and Kim, B. (2022), A Multi-Class Data Description Based Method for Classifying Predefined Defect Patterns and Detecting New Defect Patterns of Wafer Bin Maps, *Korean Institute of Industrial Engineers*, **48**(3), 298-309.
- Su, T. and Dy, J. G. (2007), In Search of Deterministic Methods for Initializing K-means and Gaussian Mixture Clustering, *Intelligent Data Analysis*, **11**(4), 319-338.
- Tarjan, R. (1972), Depth-First Search and Linear Graph Algorithms, *SIAM on Computing*, **1**(2), 146-160.
- Yoo Y., Ahn D., Park S. H., and Baek J. G. (2015), Wafer Map Image Analysis Methodology in Semiconductor Process, *Journal of Industrial Engineering*, **41**(3), 267-274.
- Yu, J. and 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.
- Yu, N., Xu, Q., and Wang, H. (2019), Wafer Defect Pattern Recognition and Analysis Based on Convolutional Neural Network, *IEEE Transactions on Semiconductor Manufacturing*, **32**(4), 566-573.
- Zhou, Y. and Chen, W. (2019), Adaptive Noise-Robust Back-Propagation Algorithm for Reconstructing Defective Sensor Images, *Sensors*, **19**(7), 1712.

저자소개

최이수 : 한양대학교 산업경영공학과에서 석사학위를 취득하고, LG에너지솔루션 구동부예지보전개발팀에서 근무 중이다. 주요 관심 분야는 머신러닝, 데이터마이닝이다.

윤주호 : 명지전문대학 산업시스템경영공학과에서 학사학위를 취득하고, 한양대학교에서 산업경영공학과 석박사통합과정에 재학 중이다. 주요 관심분야는 통계 학습, 기계 학습, 데이터마이닝, 기술경영이다.

김병훈 : 려거스 대학교에서 산업시스템공학 박사학위를 취득하고, 현재 한양대학교 산업경영공학과에 재직 중이다. 주요 관심분야는 통계 데이터 마이닝 방법론 개발, 반도체 제조 공정용 데이터 마이닝 모델 개발, 그래프 마이닝이다.