

액티브 러닝을 활용한 반도체 웨이퍼 신규 불량 패턴 검출

황성진 · 백인성 · 김성범[†]

고려대학교 산업경영공학과

Active Learning for Detecting New Defect Patterns in Wafer Bin Maps

Sung Jin Hwang · Insung Baek · Seoung Bum Kim

Department of Industrial and Management Engineering, Korea University

In the semiconductor manufacturing industry, a wafer bin map (WBM) contains defect patterns that provides important clues to identify the root causes of the defect. Traditionally, field engineers classify the pattern types by manually checking WBM. Recently, many studies have been conducted for automatic classification by using deep learning models. To accurately classify defect patterns with convolutional neural network (CNN)-based deep learning models, every WBM must have accurate pattern labels. However, in reality, it takes a lot of time and efforts for engineers to label all the data. In addition, existing CNN-based studies show limitations that cannot detect new defect patterns, frequently occurred in real situations. In this study, we devise a new pattern detection framework based on active learning. Through this, new patterns can be selectively detected even with existing active learning methodologies, and classification performance can be secured by effectively sampling unlabeled data. And we compared the performance and characteristics of each sampling strategy for new pattern detection. The usefulness and applicability of this study was demonstrated by WM-811K, publicly available WBM data.

Keywords: Semiconductor Manufacturing, Defect Identification, Wafer Bin Map, Active Learning, Sampling Strategy, Convolutional Neural Network.

1. 서론

반도체 제조 산업의 핵심인 웨이퍼는 반도체를 만들기 위한 바탕이 되는 원판으로, 한 장의 웨이퍼 안에 수백 개의 칩을 반복적으로 나열하여 공정을 진행한다. 반도체 제조 공정은 노광(photo), 식각(etching), 확산(diffusion), 박막증착(chemical vapor deposition), 금속배선(metalization), 평탄화(chemical mechanical polishing), 세정(clean), 이온주입(ion implantation)의 8대 공정의 조합으로 이루어져 있으며, 웨이퍼 단위로 수백 개의 복잡한 공정 단계를 거쳐 완성된다. 이 과정 중에 제품에 불량이 발생하면 제품의 품질과 수율을 떨어뜨리고, 이는 기업의 이윤과 직결되기 때문에 발생한 불량에 대한 빠른 원인 파악 및 조치는 반도체 제조

공정의 가장 중요한 문제 중 하나이다. 이 문제에 대해 Chen *et al.*(2005)은 제조 공정 중 거처간 설비 이력에 따라 변동되는 웨이퍼의 수율을 이용하여, 룰 마이닝 기반의 불량 설비 탐색 방법을 제시하였으나, 설비 진행 이력 외에도 구체적인 불량의 원인을 파악하기 위해서는 다양한 검사를 통한 현상 확인이 필요하다. 대표적으로 제조 공정이 완료되면 완성된 웨이퍼 위의 반도체칩이 정상적으로 작동하는지 확인하는 electrical die sorting(EDS) 검사를 통해 수율과 불량을 확인할 수 있다. 모든 웨이퍼는 EDS 검사를 진행하며, 이 EDS 검사의 결과 데이터로 웨이퍼 빈 맵(wafer bin map; WBM)이 생성된다. WBM은 각 칩이 어떤 종류의 불량을 가지고 있는지 칩의 위치 정보와 함께 보여주기 때문에, 불량의 원인에 따라 특정한 불량 패턴을 가진다. 이 때문에

[†] 연락저자 : 김성범 교수, 02841 서울특별시 성북구 안암로 145, 고려대학교 산업경영공학과, Tel : 02-3290-3397, Fax : 02-929-5888, E-mail : sbkim1@korea.ac.kr

2021년 11월 9일 접수; 2022년 1월 12일 수정본 접수; 2022년 2월 8일 게재 확정.

<Figure 1>과 같이 WBM은 엔지니어가 불량률의 원인을 분석하는데 중요한 단서를 제공한다. 즉, 동일한 형태의 불량 패턴을 갖는 웨이퍼는 그 불량률의 유발 원인이 동일할 것임을 추측할 수 있으며, 특정한 패턴이 일정 기간 내에 다수 발생한다면 엔지니어는 이를 통해 불량률의 원인을 파악하고 조치할 수 있을 것이다(Hsu and Chien, 2007). 이 때문에 WBM 패턴을 분류하는 문제는 매우 중요하며 현업에서는 숙련된 엔지니어에 의해 수행되고 있다. 다만, 기업은 이를 보다 효율화하기 위해 향상된 분류 속도와 일정 수준 이상의 성능을 가진 자동분류 시스템을 필요로 하고 있다. 이러한 이유로 머신러닝 방법론을 통해 이를 해결하고자 하는 다양한 시도들이 진행되고 있다.

머신러닝 모델이 WBM을 분류하기 위해서는 먼저 맵 이미지로부터 특징을 추출하는 과정이 필요하며, 추출한 특징을 기반으로 불량 패턴을 분류하는 딥러닝 신경망 모델이 최근까지 다수 연구되었다. 이러한 방법들은 분류 문제에 대해서는 효과적으로 작동할 수 있으나, 실제 산업 현장에서는 다음과 같은 추가적인 문제점을 안고 있다.

- 반도체 제조 산업에서는 매일 수만 매 이상 검사 데이터가 만들어진다. 이러한 데이터들은 전부 레이블이 없는 상태로, 엔지니어에 의해 분류작업이 이뤄지며 이는 시간과 비용을 크게 요구한다. 따라서 이러한 대량의 레이블이 없는(unlabeled) 데이터에 대한 분류 비용을 최소화할 수 있어야 한다.
- 일반적인 상황에서는 기존에 경험한 불량 패턴이 반복적으로 발생하지만, 간헐적으로 신규 패턴이 발생한다. 신

규 패턴이란 기존 패턴과는 형태가 달라서, 사용하던 레이블로는 분류할 수 없고, 새로운 이름의 분류가 필요한 패턴이다. 신규 패턴은 기존에 분석된 경험 데이터가 없기 때문에, 새롭게 발생 원인을 찾고 조치해야 한다. 만약 신규 패턴의 검출과 분석이 늦어진다면 이는 품질 사고로 이어질 수 있으므로, 엔지니어가 이를 빠르게 인지할 수 있도록 모델이 신규 패턴을 구분할 수 있어야 한다.

- <Figure 2>와 같이 신규 패턴도 여러 종류가 될 수 있기 때문에, “하나의” 신규 클래스로 분류되는 것이 아니라, 유형별로 각각 새롭게 추가할 수 있어야 한다. 불량률의 원인 파악을 위해서는 세부적인 유형 분류가 필요하며, 모델이 추가된 신규 패턴 종류별로 분류 성능을 획득할 수 있어야 한다.

이러한 상황에 준지도학습의 일종인 액티브 러닝(active learning)이 효과적으로 사용될 수 있다. 액티브 러닝은 대량의 unlabeled 데이터 중에서 성능 향상에 가장 효과적일 것으로 예상되는 샘플을 선별 레이블링하여 학습하는 방법론으로, 모델과 엔지니어가 협력한다는 특징을 가지고 있다. 특히 하루에도 수만 장이 생성되는 웨이퍼 데이터를 모두 분류하기 어려운 산업 현장에서 레이블링 비용을 줄이면서 모델의 성능을 획득하는데 효과적인 방법으로 사용될 수 있다. 액티브 러닝이 작동하는 순서는 <Figure 3>과 같다. 먼저 소량의 레이블이 있는 데이터를 가지고 모델을 구축한다. 이렇게 구축된 모델은 다시 unlabeled 데이터의 레이블을 예측하면서, 앞으로 성능 향상에 도움을 줄 수 있는 샘플을 골라내어 엔지니어에게 제공한다.

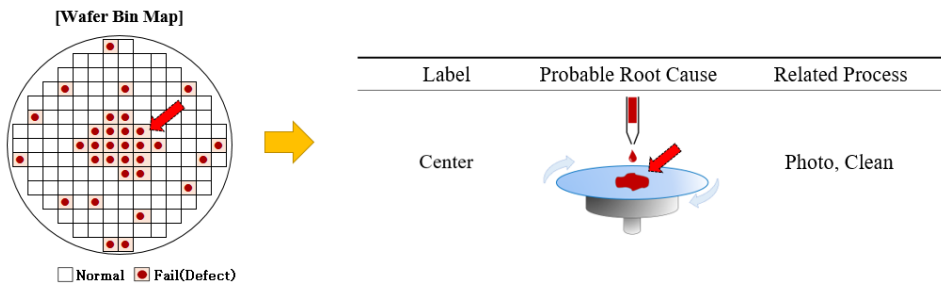


Figure 1. Wafer Bin Map and Example of Root Cause (e.g., center map is caused by the process of applying the solution in the center, such as photo or clean process)

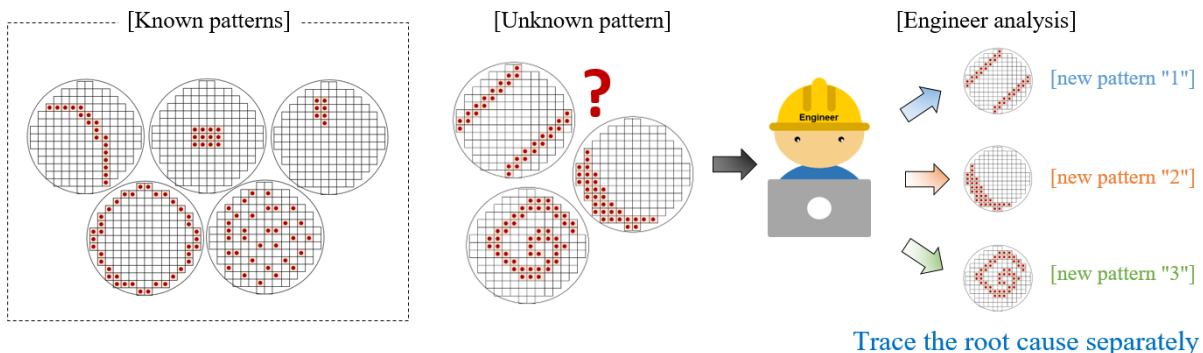


Figure 2. An Example of Unknown Patterns Occurred in a Process

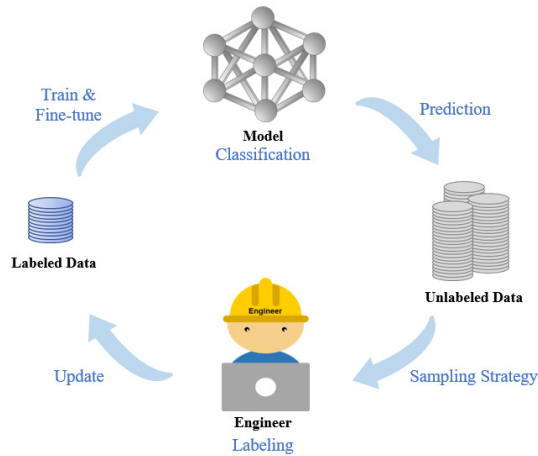


Figure 3. The General Process of Active Learning

여기서 어떤 샘플이 성능에 도움이 될지 계산하여 샘플링하는 단계를 “샘플링 전략(sampling strategy)”이라고 표현하며, 액티브 러닝의 가장 핵심적인 단계이다. 엔지니어는 이 샘플을 레이블링하여 기존 학습 데이터에 추가한 뒤 모델을 재학습 시키면, 모델은 양질의 학습 데이터의 증가로 인해 성능을 높일 수 있다. 이 과정을 반복하면, 단순히 불특정샘플을 추가하는 것보다 훨씬 효율적으로 일정 수준 이상의 성능을 얻을 수 있기 때문에 비용을 절약하는 효과적인 방법으로 사용될 수 있다. Shim *et al.*(2020)은 WBM의 분류 문제에 액티브 러닝을 적용하여 좋은 성능을 보였으나, 이러한 기존 제안 방법론들은 신규 패턴의 발생을 배제하고 기존에 알고 있는 패턴 종류에 대해서만 예측하였다. 하지만 실제 반도체 산업 현장에서는 일정한 분류 성능 이상으로 신규 패턴에 대한 감지가 굉장히 중요한 요소이다. 이에 본 연구에서는 신규 패턴 감지에 유효한 액티브 러닝 프레임 워크를 구성하고, 이를 웨이퍼 빈 맵 분류 문제에 적용하여 레이블링 비용을 줄이면서도, 동시에 신규 패턴을 조기에 검출할 수 있으며, 최종적으로 학습된 모델이 신규 패턴을 포함해서 좋은 분류 성능을 보였음을 확인하였다. 본 논문의 주요 기여점은 다음과 같다.

- 웨이퍼 빈 맵 분류 문제에 액티브 러닝을 적용하여 신규 패턴을 가진 웨이퍼를 검출하고, 이를 훈련 데이터에 추가했을 때 모델이 학습하여 분류할 수 있는 프레임 워크를 구성하였다.
- 실제 웨이퍼 빈 맵 데이터에 적용하여 신규 패턴에 대한 분류 성능을 단기간에 획득하였으며, 이를 통해 기존의 샘플링 전략만으로도 성능 향상뿐만 아니라, 신규 패턴을 조기 검출할 수 있음을 보였다.
- 신규 패턴 검출 문제에 대해 다양한 샘플링 전략을 적용하여 성능을 비교하였으며, 각각의 특징 및 장단점을 분석하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 WBM 분류 문제에 대한 기존 연구를 설명하였다. 제3장에서는 사용하는 데이터와 액티브 러닝 프레임 워크에 대해 설명하였다. 제4장에

서는 WBM 분류 문제에 대해 적용했을 때, 모델의 성능 결과를 정리하였다. 끝으로 제5장에서는 본 연구의 결론 및 기대효과를 제시하였다.

2. 관련 연구

2.1 Wafer Bin Map Classification

WBM 분류 문제는 반도체 산업의 성장과 더불어 효과적인 분류방법론들에 대한 다양한 시도들이 행해져 왔다. 초기에는 웨이퍼의 특징을 추출하기 위한 통계적 분석이 이뤄졌으나 이는 웨이퍼가 가진 칩의 위치정보와 불량 패턴을 정확하게 나타내기에는 한계가 있었다(Hsu and Chein, 2007). 때문에 머신러닝의 발달과 함께 다양한 알고리즘을 통한 분류가 시도되었다. Wu *et al.*(2015)은 본 연구에도 동일하게 사용된 웨이퍼 빈 맵 오픈데이터 “WM-811K”를 배포하고, 이 데이터셋에 대해 support vector machine(SVM)을 적용하여 분류하였다. Piao *et al.*(2018)은 웨이퍼의 특징을 라돈 변환(radon transform)을 통해서 추출하고, 다중 의사결정나무(decision tree) 앙상블 방법론을 통해 분류하였다. 하지만 이러한 방법들은 웨이퍼의 집적도가 높아짐에 따라 국부 영역에서 발생하는 미세 패턴에 대한 분류 정확도가 떨어지는 단점이 있다. 이에 최근에는 합성곱 신경망(convolutional neural network, CNN)을 이용한 연구가 진행되고 있다. Lee *et al.*(2017)은 CNN 구조를 사용하여 효과적으로 웨이퍼 불량 패턴을 감지하였고, Nakazawa and Kulkarni(2018)는 특수한 패턴까지 포함한 실제 웨이퍼 데이터에 대해서도 CNN을 통해 좋은 성능을 보였다. Kyoung and Kim(2018)은 웨이퍼 위에 하나가 아닌 여러 패턴이 섞인 불량에 대해서 분류하였다. 이러한 연구들은 지도 학습이기 때문에 레이블이 있는 데이터가 충분히 제공된다면 굉장히 좋은 성능을 발휘할 수 있다. 하지만 정확한 레이블이 있는 데이터를 실제 산업 현장에서 얻기 위해서는 큰 비용이 발생한다는 문제점이 있어, 레이블이 없는 데이터를 다룰 수 있는 방법론이 필요한 실정이다. 이러한 문제에 대해 Alawieh *et al.*(2018)은 “pseudo-boolean satisfiability solver”를 사용하여 웨이퍼의 특징을 나타낼 수 있는 최소단위로 군집화한 뒤, 이를 엔지니어가 확인할 수 있는 비지도 학습을 고안하였다. 하지만 이 경우 군집 내에서도 세부적인 분류가 필요할 때, 이를 반영한 좋은 성능을 내기에는 한계점이 있었다. 준지도학습이 이 문제에 대한 좋은 방법론이 될 수 있으며, Kahng and Kim(2021)은 pretext invariant representation learning(PIRL)이 적용된 CNN 기반 준지도 학습 모델을 통해 적은 수의 레이블 데이터로부터 전체 데이터를 분류할 수 있도록 하였다. Shim *et al.*(2020)은 이 문제에 대해 액티브 러닝을 적용하여 소량의 레이블링만으로도 충분히 좋은 성능을 보였다. 다만 산업 현장에서는 준지도 학습을 통해 레이블이 없는 데이터를 분류하는 것이 상으로, 계속해서 발생하는 신규 패턴에 대해서 모델이 이를

인지하고, 새로운 패턴에 대해 빠르게 학습하는 태스크가 무엇보다 중요할 수 있다. 앞서 연구된 준지도학습은 모델에 새로운 패턴이 추가될 때, 이를 모델이 인지하기 어렵다는 문제점을 안고 있다. 따라서 본 연구는 이러한 신규 패턴을 인지할 수 있는 액티브 러닝 구조를 구성하고, 새롭게 학습한 신규 패턴에 대한 분류를 포함해서 모델이 조기에 충분한 분류 성능을 획득할 수 있도록 개선하고자 한다.

2.2 Active Learning

액티브 러닝은 레이블링 비용이 큰 경우에, 적은 수의 샘플 만으로도 모델의 성능을 극대화시킬 수 있는 방법론이다. 모델이 샘플을 골라내면, 숙련된 엔지니어에 의해 레이블링 되어 학습에 사용한다. 어떤 샘플이 모델의 성능을 극대화시킬 수 있는가에 대한 접근 방식은 매우 다양한데, 주로 현재의 모델이 추출한 특징 벡터로는 정확하게 분류하기 어려운 샘플일 때 이를 훈련 데이터로 추가하면 모델의 성능을 높일 수 있다고 보는 것이 대표적이다.

가장 먼저 확률적인 접근은 특정 클래스로 분류될 확률이 크지 않은 샘플을 불확실한 샘플로 보고 이를 훈련 데이터로 추가하는 방식이다. 이러한 불확실성을 계산하는 방식은 고전적으로 세 가지가 있다. Settles and Craven(2008)은 클래스별 확률 중에서 가장 높은 확률의 값이 작을수록 불확실성이 큰 샘플로 판정하는 최소 신뢰(least confidence)를 제안하였고, Scheffer *et al.*(2001)은 확률이 가장 큰 첫 번째, 두 번째 클래스 간의 확률 차이를 이용하는 margin을 제안하였다. Hwa(2004)는 엔트로피를 이용하여 불확실성을 계산하는 방식을 사용하였다. 앙상블 접근은 여러 가지 모델이 특정 샘플에 대해 예측했을 때, 그 결과값이 서로 상이한 샘플이 판단하기 어렵고, 성능향상에 도움이 되는 샘플이라고 접근했다. Seung *et al.*(1992)는 “query by committee”를 통해 각 모델이 서로 다른 예측을 하는 샘플을 선택하여 성능 향상을 꾀했다. 베이지안 접근은 베이지안 신경망의 불확실성을 곧 샘플링의 기준으로 둔다. Gal and Ghahramani(2016)는 베이지안 신경망의 불확실성을 드롭아웃(drop out)을 통해 추정하고, 이를 통해 샘플링하는 방법을 제안하였다. 분포 접근은 데이터의 분포를 통해 샘플링하는 방식이며, Sener *et al.*(2018)은 “core set”을 제안하여, 불확실성을 계산하는 것이 아니라, 훈련 데이터의 분포를 전부

계산하여, 샘플 간 거리를 통해 나머지 샘플을 대표할 수 있는 중심 샘플을 계산하였다. Yoo and Kweon(2019)은 모델의 중간 레이어의 특징 벡터를 입력받아 모델 자체의 손실값을 추정하는 별도의 모델을 추가하여, 예측되는 손실값이 큰 샘플을 골라 좋은 성능을 얻었다.

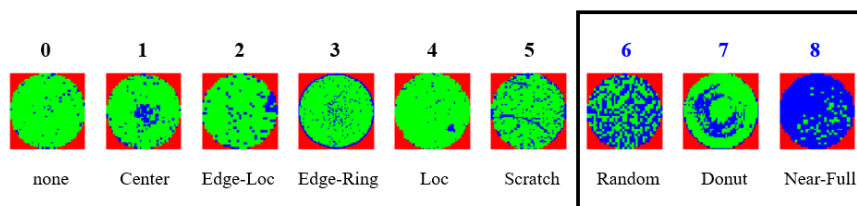
즉 어떤 샘플링 방법론이라도, 모델이 임의의 샘플을 업데이트하는 것보다 샘플링 전략에 의해 선택된 샘플을 학습할 때, 더 좋은 성능을 낼 수 있었다. 그러나 위 연구들은 샘플링 전략 자체에 대한 연구로, 실제 상황에서 종종 발생하는 “경험하지 못한 신규 클래스”가 있는 경우를 고려하지 않았다. 따라서 액티브 러닝 과정 중간에서 엔지니어가 직접 패턴을 구분해서 레이블링할 수 있는 이점을 이용하여, 신규 패턴을 조기에 검출하면서 모델이 다양한 신규 패턴을 지속적으로 추가 분류할 수 있도록 학습할 수 있도록 개선하고자 한다.

3. Active Learning for Detecting New Patterns

본 연구에서는 액티브 러닝의 샘플링 전략이 신규 패턴을 검출했을 때, 추가된 신규 패턴 클래스의 개수에 따라 CNN 모델이 유연하게 출력값을 변경하면서 학습할 수 있는 프레임워크를 사용한다. 또한 이러한 구조를 통해 기존의 샘플링 전략 만으로도 효과적으로 신규 패턴을 조기에 검출할 수 있음을 확인하고, 전략에 따른 검출 성능을 비교할 것이다. 이를 확인하기 위해 신규 패턴 검출이 필요한 문제 상황을 조성하였으며 본 장에서는 실험 환경과 사용된 프레임워크에 대해 설명한다. 먼저 WBM 데이터의 전처리 과정과 모델의 전체적인 구조를 설명하고, 세부적으로 사용된 CNN 모델, 그리고 사용된 액티브 러닝의 샘플링 전략에 대해 각 단계별로 설명하였다.

3.1 Data Preprocessing

실험에 사용한 WM-811K 데이터셋은 811,457매의 실제 웨이퍼 빈 맵을 가지고 있는 오픈데이터이다(<https://www.kaggle.com/qingyi/wm811k-wafer-map>). 이 중에서 불량 패턴에 따라 레이블이 표기된 웨이퍼는 172,950매고, 총 9개 클래스를 가지고 있다. 그 패턴에 따라 None, Center, Edge-Loc, Edge-Ring, Loc, Scratch, Random, Donut, Near-Full 이 있으며, <Figure 4.>는



3 patterns are considered as "new patterns."

Figure 4. The Example of Defect Patterns in WM-811K Wafer Bin Maps

Table 1. Description of a data set. The initial labeled data were divided into 1% of the total, and “new class” was not included. 20% of the total data were used separately for the test data

Defect Patterns	Total	Initial Labeled (1%)	Unlabeled	Test (20%)
None	147,431	1,474	116,765	29,192
Edge-Ring	9,680	97	7,666	1,917
Edge-Loc	5,189	52	4,109	1,028
Center	4,294	43	3,401	850
Loc	3,593	36	2,846	711
Scratch	1,193	12	945	236
Random	866	-	693	173
Donut	555		444	111
Near-Full	149		119	30
Total	172,950	1,714	136,988	34,248

각 패턴들의 예시를 보여준다. 본 연구에서는 레이블이 있는 172,950매를 데이터로 사용하였다. 데이터의 전처리는 먼저, 각 웨이퍼의 크기가 64×64 로 통일되도록 스케일링하였으며, 각 픽셀별 값이 0:Wafer가 없는 영역, 1:양호Chip, 2:불량Chip으로 나뉘는 것을 범주형의 one-hot 벡터 형태인 3개 채널로 사용하여 최종적으로 $[64 \times 64 \times 3]$ 차원의 이미지 형태로 만들어 주었다.

먼저 전체의 20%는 테스트셋으로 분리하고, 나머지 80%는 전부 표기된 레이블을 가려서 unlabeled 데이터로 만들어 실제 제조 현장과 최대한 동일한 상황으로 구현하였다. 모델을 구축하기 위해 전체의 1%수준(1,714매)만을 레이블링하여 사용하였으며, 이 때 액티브 러닝에서의 사람이 레이블링하는 과정을, 본 연구에서는 가려진 레이블을 다시 가져오는 것으로 대체하여 진행하였다. 본 연구는 unlabeled 데이터 내에 신규

패턴이 존재할 경우, 모델이 이를 검출하여 분류할 수 있도록 구성하기 위해 처음 구축된 모델이 신규 패턴의 존재를 알 수 없도록 하였다. 이를 위해 <Table 1>과 같이 최초 모델 구축을 위한 훈련데이터에서 데이터의 개수가 가장 적은 Random, Donut, Near-Full 세 가지 패턴을 제외시켰다. 따라서 모델은 최초 구축 시에는 6개 클래스만 예측 가능하며, 대량의 unlabeled 데이터에만 모델이 알지 못하는 신규 패턴이 추가로 존재하게 된다. 액티브 러닝이 진행되며 샘플링될 때, 최초 6개 클래스 예측으로 시작해서 신규 패턴을 효과적으로 선별하여 학습하면, 최종적으로 신규 패턴을 포함하여 9개 클래스 모두를 분류할 수 있는지 확인하였다.

3.2 Active Learning Structure Overview

기존 액티브 러닝 구조와 달리 신규 패턴을 검출하여 이를 모델이 학습할 수 있도록 만들기 위해, 훈련 데이터의 클래스의 개수에 따라 모델의 출력력이 조정될 수 있도록 변형된 구조를 사용하였다. <Figure 5>와 같이 먼저 레이블링된 초기 훈련 데이터를 통해 모델이 구축되고 나면, unlabeled 데이터를 모델에 입력하여, 계산된 출력값을 샘플링 전략에 사용한다. 샘플링 전략은 unlabeled 샘플 중에서, 모델이 분류하기 어렵거나, 기존의 분류에 적합하지 않은 샘플로 k 개를 뽑도록 계산된다. 본 연구에서는 100개씩($k=100$) 샘플링하여 엔지니어에게 제공되도록 하였으며, 이는 엔지니어에 의해 레이블링 처리되어 훈련 데이터로 업데이트된다. “모델 학습 → 샘플링 전략 → 엔지니어 레이블링 → 훈련 데이터 업데이트”의 단계를 묶어 1 phase라고 하며, 액티브 러닝은 phase를 반복할 때마다 훈련 데이터의 개수가 초기(1,714매)+ $100 \times n$ phase만큼 늘어나고 이를 통해 성능을 향상시킨다. 이전의 연구가 모든 class의 존재를 미리 알고 시작한 상태로 성능을 향상시켰다면, 본 연구는 알고 있는 패턴에 대한 분류 성능만으로도, 샘플링 전략에

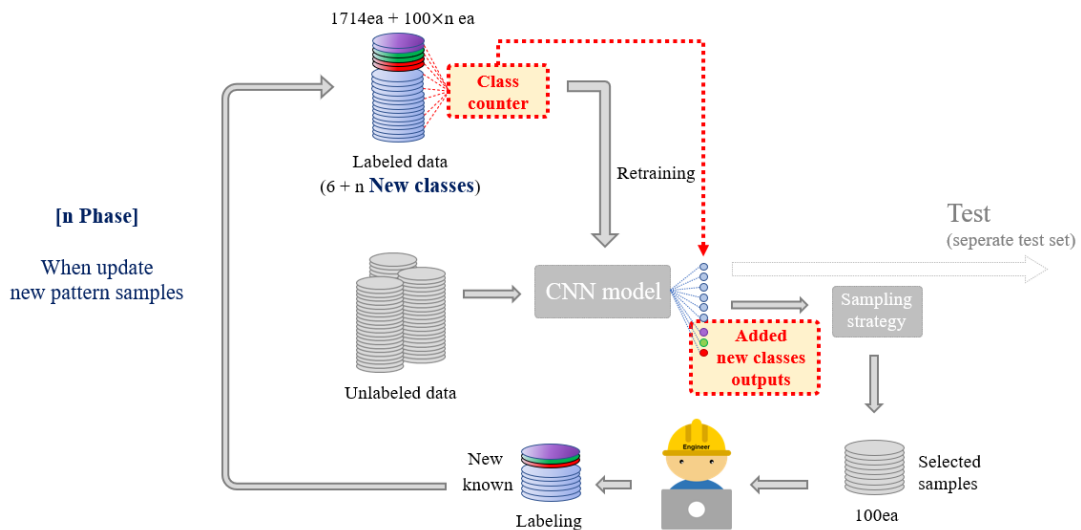


Figure 5. Active Learning Structure for Improving Detection Ability of New Pattern

의해 현재 분류와는 다른 특징 벡터를 골라낼 수 있기 때문에 신규 패턴이 감지될 수 있을 것이라고 기대한다. 즉, 액티브 러닝의 샘플링 전략이 가진 특성에 의해서 unlabeled 데이터 내의 신규 패턴 샘플을 우선적으로 선별하여 검출할 수 있으며, 엔지니어는 이를 확인하여 신규 클래스로 분류하여 훈련 데이터에 업데이트하면 모델은 재학습을 통해 신규 패턴에 대한 성능을 얻을 수 있다. 이는 엔지니어가 대량의 데이터를 일일이 탐색하여 레이블링하지 않고도, 모델이 찾아내는 신규 패턴을 우선적으로 학습시킬 수 있기 때문에, 레이블링 비용을 줄일 수 있다는 의미를 가진다. 이를 구조적으로는 훈련 데이터가 업데이트될 때 클래스 카운터가 신규 클래스의 여부를 확인하고, 이를 통해 모델이 새로운 클래스의 종류 n 개를 추가하여 출력 클래스의 개수가 기존 6개에서 $6+n$ 개로 조정되도록 구현하였다. 이 때, 모델의 출력 클래스의 개수가 변하더라도, 해당 phase까지 학습해온 표현(representation)을 잃지 않도록, 늘어난 클래스에 맞춰 추가 학습만 진행되게 가중치(weight)를 유지시켰다. 조정된 출력값을 갖게 된 모델은 이후에도 계속해서 phase를 반복하며, 샘플링 전략을 통해 훈련 데이터를 업데이트하여 성능을 향상시킨다. 최종적으로 모델은 본 실험에서 가정한 신규 패턴을 포함한 9개 전체 클래스에 대해 전부 일정 수준 이상의 성능을 확보할 수 있다.

3.3 Convolution Neural Network Classification Model

액티브 러닝을 위해서는, 대량의 unlabeled 데이터에 대해 반복적으로 계산해야 하므로 모델의 복잡도를 줄여 효율성을 극대화하는 방향으로 설계하였다. 웨이퍼의 특징을 추출하기 위해 CNN을 기반으로 사용되던 모델 중 비교적 가벼운 모델을 사용하였으며, 전체 구조는 <Figure 6>과 같다. 총 4개의 convolutional layer를 사용하였으며, kernel size는 3×3을 사용하였고, 각각 64, 128, 256, 512 채널을 갖는다. 각 층마다 활성화 함수로는 rectified linear unit(ReLU)을 사용하였으며, 2×2의

맥스 풀링을 통해 크기를 반으로 줄이며 진행하고 마지막단에는 fully-connected layer 2개를 사용하여 256, 128 차원으로 특징 벡터를 추출하였다. 최종적으로 각 클래스별 확률을 구하고 소프트맥스를 사용해 분류하였다. 이 때, 모델의 마지막 출력은 신규 패턴이 추가됨에 따라 그에 맞춰 유연하게 변형될 수 있도록 구성하였다. 특히 클래스의 개수가 변경될 때, 이전 phase까지 학습해온 모델의 가중치를 잃지 않도록 첫 convolutional layer부터 두 번째 fully-connected layer(128)까지의 가중치를 저장 및 불러오면서 모델의 성능이 유지되도록 하였다.

3.4 Sampling Strategy

본 연구의 핵심은 액티브 러닝의 샘플링 전략이 모델의 성능을 올리는 샘플을 골라낼 뿐 아니라, 신규 패턴을 감지하여 엔지니어에게 제공할 수 있다는 점이다. 따라서 본 장에서는 어떻게 샘플링 전략이 다양한 계산을 통해 신규 패턴을 샘플링할 수 있는지 구체적으로 설명한다.

본 연구에는 효율적으로 샘플링하기 위한 세가지 접근 방식과 여기에 속하는 다섯 가지 샘플링 전략에 대해 구체적인 계산 방법론을 사용하였다. 확률적 접근, 베이지안 접근, 거리 접근의 세 가지 접근 방식 모두 샘플에 대한 모델의 예측이 불확실할수록 판단하기 어려운 샘플이며, 이러한 샘플이 곧 성능에 도움이 된다는 가정을 가지고 있다. 때문에 이러한 “불확실성”을 측정하는 방법에 대해 세부적으로 설명한다. 먼저 확률적 접근은 가장 고전적인 방식으로 샘플이 어떠한 분류에도 속할 확률이 높지 않다면, 이는 현재의 모델이 판단하기 어려운 모델이라는 의미의 접근이다(Settles Settles and Craven., 2008). 이러한 접근 방식에는 대표적으로 3가지가 있다.

최소 신뢰(least confidence) 샘플링은 샘플이 갖는 확률 중에 가장 큰 값을 신뢰값(confidence)으로 정의하고, 이 값이 작을수록 불확실성이 높다고 판단하는 방식이다. 즉 어떤 분류로 도 속하기 어려운 샘플일수록 낮은 신뢰값을 가질 것이고, 이

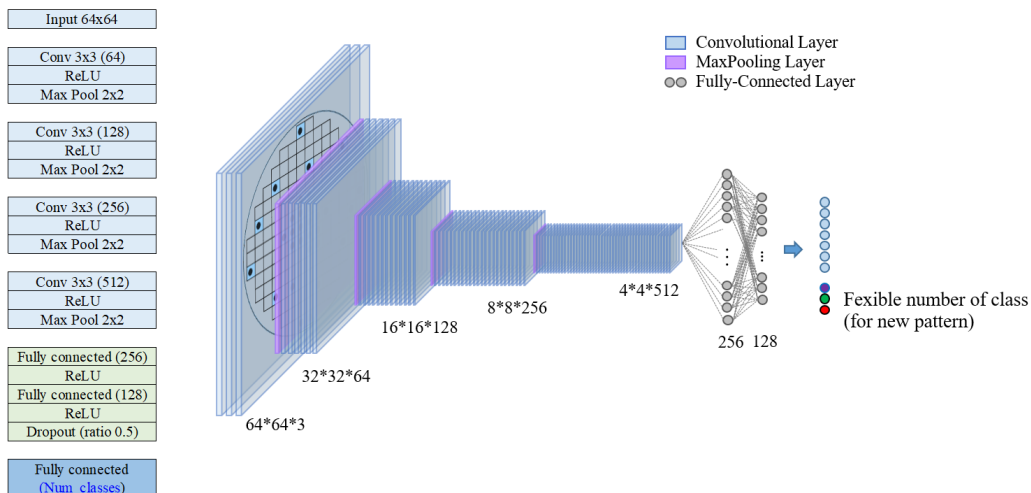


Figure 6. WBM Classification CNN Model for Learning New Patterns

를 통해 샘플을 선택할 경우 기존의 경험하지 못한 신규 패턴이 선택될 가능성이 높아진다. 최소 신뢰값 lc_i 는 i 번째 샘플의 입력 데이터 X_i 로부터 모델이 예측한 j 개 클래스의 확률 $p_j(X_i)$ 중에서 가장 큰 값이며, 식 (1)로 표현된다. 이 샘플링은 lc_i 값이 작은 샘플들을 선택한다.

$$lc_i = -\max_j p_j(X_i) \quad (1)$$

최소 확률차(least margin) 샘플링은 샘플이 갖는 확률 중에 가장 큰 값과, 두 번째 큰 값의 차이가 작을수록 선택하는 방식이다. 최소 신뢰 샘플링에서 오직 가장 높은 확률값만을 고려하기 때문에 이를 보완하기 위해 제안되었다. 최소 확률차 lm_i 는 i 번째 샘플의 예측 확률 중 가장 큰 값 $p_{j_1}(X_i)$ 과 두 번째로 큰 값인 $p_{j_2}(X_i)$ 의 차이로 계산되며, 식 (2)로 표현된다. 이 샘플링은 lm_i 가 작은 샘플을 선택한다.

$$lm_i = -(p_{j_1}(X_i) - p_{j_2}(X_i)) \quad (2)$$

엔트로피(entropy) 샘플링은 분류 확률값을 통해 계산한 엔트로피로 불확실성을 정의하며, 엔트로피가 가장 높은 샘플을 불확실성이 높다고 판단하여 선택한다. 엔트로피 en_i 은 클래스의 개수가 C 개일 때, 각 클래스별 확률 $p_j(X_i)$ 와 여기에 \log 를 취해준 값의 곱들을 모두 합하여 사용하며, 식 (3)으로 표현된다.

$$en_i = -\sum_{j=1}^C p_j(X_i) \log p_j(X_i) \quad (3)$$

베이저안 샘플링은 베이저안 신경망 모델을 사용하여 불확실성을 계산하는 방식으로, 본 연구에서는 불일치를 통한 베이저안 동적 학습(Bayesian active learning by disagreement, BALD)을 사용하였다. 베이저안 신경망에서의 불확실성 값이 큰 샘플일수록 모델이 판단하기 어려워면서, 훈련 데이터로 학습할 시 성능에 도움이 되는 샘플로 판단한다. 이 불확실성은 딥러닝 모델의 일부 노드들을 확률적으로 배제시키는 드롭아웃(dropout)을 반복적으로 수행했을 때, 변화하는 출력값의 분포를 통해 추정할 수 있다(Gal et al., 2016). 모델로 한 샘플에 대해 N 회 반복 예측할 때마다, 드롭아웃에 의해 배제되는 노드들이 바뀌면서 서로 다른 N 개의 예측 확률을 얻는다. 이 N 개의 예측 확률의 평균으로 구한 엔트로피가 pe_i 이며, 이는 식(4)로 표현된다. n 번째(i.e., $n \in \{1, \dots, N\}$) 예측 확률값의 엔트로피가 en_n 라고 할 때, 베이저안 신경망에서의 불확실성 bd_i 는 식 (5)와 같이 pe_i 과 en_n 평균값의 차이로 계산된다(Houlsby et al., 2011).

$$pe_i = -\sum_{j=1}^C \left(\frac{1}{N} \sum_1^N p_j(X_i) \right) \log \left(\frac{1}{N} \sum_1^N p_j(X_i) \right) \quad (4)$$

$$bd_i = pe_i - \frac{1}{N} \sum_{n=1}^N (en_n) \quad (5)$$

거리 접근은 샘플들의 중심에서 가장 멀리 떨어진 샘플이 불확실성이 높다고 보는 방식으로, 본 연구에서는 대표적인 K-평균 군집화(K-means clustering)를 사용하였다. 모델로부터 출력된 샘플 별 특성 벡터(본 연구의 실험에서는 fully-connected layer(dim=128)의 output)로부터 군집화를 통해 중앙값 μ_i 을 확인하고(식 (6)), 이 중앙값에서 가장 멀리 떨어진 샘플을 선택한다. K-평균을 사용하는 것은 연산량이 많아 불리할 수 있으나, 웨이퍼 빈 맵이라는 독특한 데이터 도메인에서 특징 벡터의 거리를 통해서 샘플링 했을 때, 신규 패턴을 선택하는 양을 보고자 함이다.

$$\text{distance} = \|X_j - \mu_i\|^2 \quad (6)$$

마지막으로 다섯 가지 샘플링 전략과의 비교를 위해 엔지니어가 무작위로 unlabeled 데이터를 하나씩 확인하며 레이블링하는 상황을 무작위 샘플링(random sampling)으로 진행하였다. 동일한 양의 훈련 데이터양이라도 효율적인 샘플링 전략 없이 진행했을 때의 성능을 확인하기 위함이다.

4. 실험 결과

4.1 평가 지표

본 연구에서는 F1-score를 평가 지표로 사용하였다. 다중 클래스 분류 문제에서 널리 사용되는 지표로, 본 연구와 같이 클래스 불균형이 심한 상황에서 유효하게 사용될 수 있는 지표다. 먼저 <Table 2>는 테스트 데이터의 정답 클래스와 모델이 예측한 분류에 따라 각각의 개수를 칸마다 표시한 정오행렬(confusion matrix)이다. 예를 들어 샘플의 실제 클래스가 1이고, 모델의 예측도 1이라면, true positives에 속한다. F1-score를 구하기 위해 먼저 정밀도(precision)와 재현율(recall)을 구한다. 정밀도는 특정 클래스라고 예측한 수 중에서 실제로 그 클래스가 정답인 수를 표현하고 정오행렬의 true positives와 false positives를 통해 식 (7)과 같이 계산된다. 재현율은 실제 정답 클래스의 개수 중 몇 개를 모델이 정확히 예측했는지를 표현하며 식 (8)로 표현된다. F1-score는 정밀도와 재현율의 조화 평균을 사용하며 식 (9)로 표현된다. F1-score는 0에서 1 사이의 값을 갖고, 그 값이 클수록 모델의 분류 성능이 더 좋다는 것을 보여준다.

Table 2. Confusion Matrix

		Ground Truth	
		True	False
Predicted	Positive	True Positives	False Positives
	Negative	False Negatives	True Negatives

$$\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (7)$$

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (8)$$

$$\text{F1_Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

4.2 실험 조건

먼저 모델의 손실 함수는 크로스엔트로피(cross entropy)를 사용하였고 모델의 최적화 알고리즘은 RMSProp을 사용하였다. 알고리즘의 하이퍼 파라미터는 learning rate을 0.0005, momentum은 0.3, L2 정규화 상수는 0.0001로 사용하였다. 최초 훈련 데이터는 1,714매(6개 클래스)로 시작하여 매 phase마다 100매씩 추가하는 방식으로 총 50 phase를 진행하였다. 각 phase 별로 업데이트된 훈련 데이터를 가지고 300 epoch 학습하도록 하고, validation loss가 25번 개선된 변화가 없다면 조기 종료(early stopping) 되도록 설정하여, 각 phase 시점에서 학습에 사용할 수 있는 훈련 데이터로는 가장 높은 성능에 도달하도록 충분히 학습하였다. 학습된 모델의 성능 평가는 모든 클래스가 존재하는 테스트 데이터(전체 20%, 9개 클래스)에 대해 측정하였으며, 초기에 모델이 신규 패턴에 대해 아직 감지하지 못했을 때는 해당 클래스에 대한 예측 확률을 0으로 처리한 뒤 테스트 성능을 확인하였다. 모든 실험은 샘플링 전략마다 난수 설정을 바꿔가며 10회 반복하여 평균값을 사용하였다.

4.3 실험 결과

4.3.1 분류 성능 평가 결과

본 장에서는 액티브 러닝 구조를 사용한 기존 패턴 및 신규

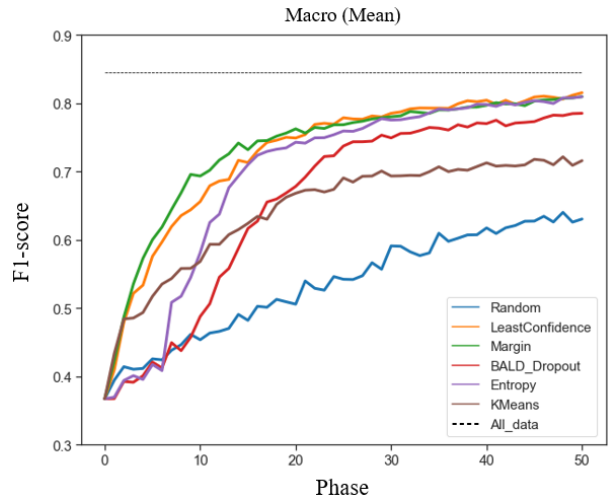


Figure 7. Comparison of F1-scores among Sampling Strategies

패턴의 분류 성능을 기술하였다. <Figure 7>은 phase별로 테스트를 진행하면서 신규 패턴까지 포함한 9개 클래스별 F1-score를 평균한 결과를 보여주고 있다. 최초 구축된 모델은 분류 성능이 부족하지만, phase를 반복하며 훈련 데이터를 쌓아 갈수록 성능이 높아짐을 확인할 수 있었다. 이 때, 무작위 샘플링은 훈련데이터가 늘어도 성능 향상이 적지만, 다른 샘플링 전략은 무작위 샘플링 대비 성능이 빠르게 향상되어 최종적으로도 더 높은 분류 성능을 보였다. 이를 통해 액티브 러닝의 phase가 진행됨에 따라 기존 패턴과 신규 패턴 분류 성능 모두를 높였음을 확인할 수 있었다.

<Table 3>은 샘플링 전략별 10 phase 간격으로 F1-score의 구체적인 값과, 괄호 안에는 10회 반복 실험한 결과의 표준 편차를 기술하였다. 무작위 샘플링의 경우 50 phase를 전부 진행해도 F1-score가 0.6306 수준에 머물렀으나, 양질의 훈련데이터를 업데이트한 샘플링 전략들은 빠르게 성능이 상승하며, 가

Table 3. Comparison F1-score of each phase of sampling strategies

Sampling	10 phase	20 phase	30 phase	40 phase	50 phase
Random	0.4537 (0.0280)	0.5059 (0.0471)	0.5912 (0.0451)	0.6176 (0.0393)	0.6306 (0.0450)
Least Confidence	0.6561 (0.0587)	0.7492 (0.0313)	0.7854 (0.0084)	0.8045 (0.0108)	0.8156 (0.0076)
Margin	0.6935 (0.0437)	0.7626 (0.0072)	0.7804 (0.0112)	0.7973 (0.0113)	0.8098 (0.0126)
Entropy	0.5812 (0.0462)	0.7431 (0.0094)	0.7756 (0.0141)	0.7984 (0.0142)	0.8097 (0.0139)
BALD_Dropout	0.4878 (0.0572)	0.6784 (0.0606)	0.7494 (0.0108)	0.7703 (0.0072)	0.7854 (0.0139)
KMeans	0.5684 (0.0433)	0.6683 (0.0371)	0.6936 (0.0072)	0.7129 (0.0097)	0.7161 (0.0140)
All_Data			0.8460 (0.0112)		

장 우수한 최소 신뢰 샘플링의 경우 50 phase에서 F1-score 0.8156에 도달하여 무작위 대비 약 0.18 더 향상된 정확도를 보였다. 이러한 실험 결과를 통해 액티브 러닝을 사용하여 기존의 샘플링 전략만으로도 신규 패턴을 더 빠르게 검출하면서도 충분한 분류 성능을 획득할 수 있음을 확인하였다. 이는 동일한 모델 조건에서 모든 unlabeled 데이터가 레이블링된 상황을 가정하여 훈련 데이터로 학습했을 때의 실험 결과인 All_data F1-score와 약 0.03의 근소한 차이로, 엔지니어에 의한 레이블링 비용을 최소화하면서도 신규 패턴을 검출할 수 있는 구조로 사용될 수 있다는 의미를 가진다.

샘플링 전략마다 어떤 샘플을 선택할 것인가에 대한 기준이 다르기 때문에 업데이트하는 샘플도 다르며, 그에 따른 성능 차이가 발생한다. 이 때문에 어떤 패턴에서 보다 큰 성능 차이가 있었는지 기준에 알고 있던 패턴과 신규 패턴으로 나누어 세부적으로 확인하였다. 먼저 <Figure 8>은 모델이 최초 구축될 시점에 기존 패턴에 대한 샘플링 전략별 F1-score 변화값이다. WM-811K 데이터셋은 심한 불균형을 가지고 있기 때문에, 다른 클래스 대비 비교적 모수가 많은 None, Center 패턴에서는 최초부터 높은 분류 성능을 보이고 샘플링 전략 간 유의차도 미약하다. 하지만 Edge-Loc, Edge-Ring, Loc 과 같은 패턴

에서는 샘플링 전략을 통해 더 높은 성능을 얻었음을 확인하였다. 특히 Scratch와 같이 수가 가장 적은 클래스 패턴의 경우, 성능의 유의차를 더욱 확연하게 보였으며, 최소 신뢰 샘플링, 엔트로피 샘플링이 가장 우수한 성능을 보였다.

<Figure 9>는 신규 패턴에 대한 정확도를 보여주고 있다. 모델이 처음 구축될 때부터 학습할 수 있었던 기존 패턴과는 달리, 신규 패턴은 액티브 러닝에 따라 업데이트된 샘플 양에 의존하기 때문에 그 성능 차이가 더욱 커진다. 신규 패턴은 unlabeled 데이터 내에서도 그 양이 매우 적어서 무작위 샘플링을 통해서도 훈련데이터로 선택될 확률이 낮다. 따라서 무작위 샘플링을 할 경우, phase를 반복적으로 진행해도 부족한 훈련 데이터량에 의해 성능이 낮음을 확인했다. 이와 달리 액티브 러닝의 샘플링 전략들은 20 phase 이내 구간에서 빠른 속도로 신규 패턴에 대한 성능향상을 보였다. 이는 샘플링 전략에 의해 감지된 신규 패턴 샘플이 훈련 데이터로 다수 업데이트되었기 때문이며, 그로 인해 높은 F1-score를 달성하였다. 이를 통해서 샘플링 전략이 신규 패턴 검출에 효과적으로 사용될 수 있으며, 단순히 샘플링을 통한 검출만이 아니라 이후에도 진행되는 phase 반복학습을 통해서 신규 패턴에 대한 정확도를 동시에 확보할 수 있음을 보였다.

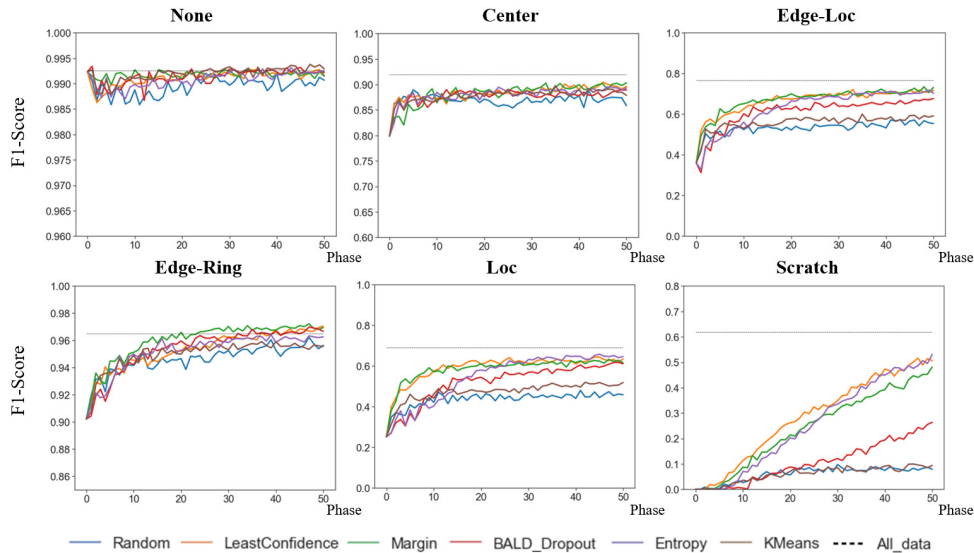


Figure 8. Comparison of sampling strategy for each “known pattern.”

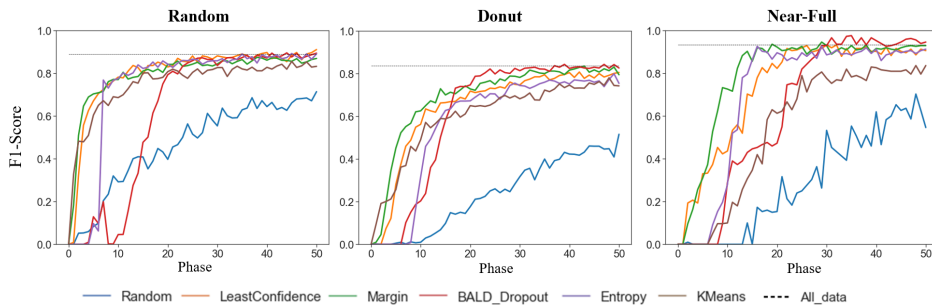


Figure 9. Comparison of sampling strategy for each “new pattern.”

또한 이러한 결과를 통해 신규 패턴의 조기 검출이라는 문제에 대한 각 샘플링 전략의 특징과 장단점을 비교할 수 있다. 먼저 확률적 접근법인 최소 신뢰, 최소 확률차, 엔트로피 샘플링의 경우 기존 패턴으로 분류될 확률값이 작은 샘플을 선택하는 특징 때문에, 초기 phase부터 기존 패턴이 아닌 신규 패턴을 빠르게 검출할 수 있다. 이는 초기 phase에서 신규 패턴에 대해 확률적 접근 전략의 성능이 가장 빠르게 상승하는 것을 통해 확인하였다. 반면 베이지안 샘플링은 드롭 아웃에 의해 변동되는 값이 큰 샘플을 우선적으로 선택하기 때문에, 초기 phase에서 신규 샘플을 검출하기보다 현재 모델이 알고 있는 패턴에 대한 분류 성능을 올리는 방향으로 작용하는 것으로 보였다. 이 때문에 10phase 이전까지 신규 패턴 검출은 비교적 느리게 진행되지만, 모든 신규 패턴에 대해 일정 이상의 성능을 획득한 이후에는 성능이 크게 증가하여 신규 패턴에 대한 최종 분류 성능은 가장 높았다. 마지막으로 KMeans 샘플링의 경우, 기존 패턴의 특징 벡터와 거리가 먼 샘플이 차례대로 선택되기 때문에 초반까지는 신규 패턴 검출을 빠르게 할 수 있지만, 가장 멀리 떨어진 신규 패턴 샘플들이 모두 확인된 이후부터는 성능의 획득량이 타 전략에 비해 부족하였다. 이는 KMeans 샘플링이 각 군집에서 거리가 먼 순서대로 선택하기 때문에, 후반부에서는 군집에서 상대적으로 멀지 않은 샘플을 업데이트하면서 성능 획득량이 감소한 것으로 보였다. 신규 패턴의 검출을 포함한 액티브 러닝 구조에 효과적으로 사용될 수 있는 샘플링 전략을 종합적으로 평가하였을 때, 신규 패턴을 빠르게 검출하고, 이후 성능 또한 꾸준히 높일 수 있는 확률적 접근법이 우수하였다.

4.3.2 훈련 데이터(Labeled data)의 클래스별 모수 비교

앞선 실험 결과에서, 모델의 분류 성능은 업데이트된 샘플에 따라 유의차를 보였다. 이에 본 장에서는 액티브 러닝이 진

행되며 phase마다 모델이 학습한 각 패턴별 훈련 데이터의 양적 변화를 확인하였다. <Figure 10>은 6개의 샘플링 전략별로 선택된 샘플이 업데이트됨에 따라 증가하는 각 클래스별 훈련 모수를 표시하였다. 최초 모델 구축 시 1,714개로 시작하여 phase마다 100개씩 업데이트 되어 데이터가 늘어나고, 이 때 샘플링 전략이 어떤 클래스의 샘플을 업데이트 했는가에 따라 비율이 달라진다. 클래스 불균형에 의해 처음에는 none 패턴이 모수의 대부분을 차지한다. 무작위 샘플링의 경우 phase가 진행되어도 최초의 클래스 비율이 그대로 유지되며, none 패턴이 훈련데이터의 대부분을 차지하고 극소수인 신규 패턴에 대한 업데이트는 거의 이뤄지지 않는다. 반면, 신규 패턴을 포함한 액티브 러닝의 샘플링 전략에 의해서는 기존 패턴에도 none이 아닌 클래스의 비율이 확연하게 증가하고, 특히 극소수의 신규 패턴도 훈련데이터로 다수 업데이트되는 것을 확인할 수 있었다. 즉 샘플링 전략이 신규 패턴 검출에 효과적임을 업데이트된 양을 통해 확인하였다.

4.3.3 신규 패턴 검출 성능 평가 결과

본 장에서는 각 샘플링 전략별 신규 패턴에 대한 검출 성능을 평가하였다. 샘플링 전략이 확보하는 신규 패턴의 매수를 검출 성능의 지표로 사용할 수 있으며, <Figure 11>에서 신규 패턴의 클래스 별로, Phase에 따라 누적되는 샘플의 매수를 보여주고 있다. 신규 패턴 샘플은 모델 구축을 위한 초기 데이터에는 존재하지 않기 때문에 0매에서 시작하고, phase가 진행되며 검출되는 만큼 훈련데이터에 업데이트된다. 더 많은 양의 샘플을 확보할수록 검출 성능이 좋다고 평가할 수 있으며, 샘플링 전략에 따라 검출되는 시점과 확보되는 양을 기준으로 비교하였다.

무작위 샘플링의 경우 phase가 진행됨에도 소수만 존재하는 신규 패턴을 거의 업데이트하지 못하는 반면 샘플링 전략에

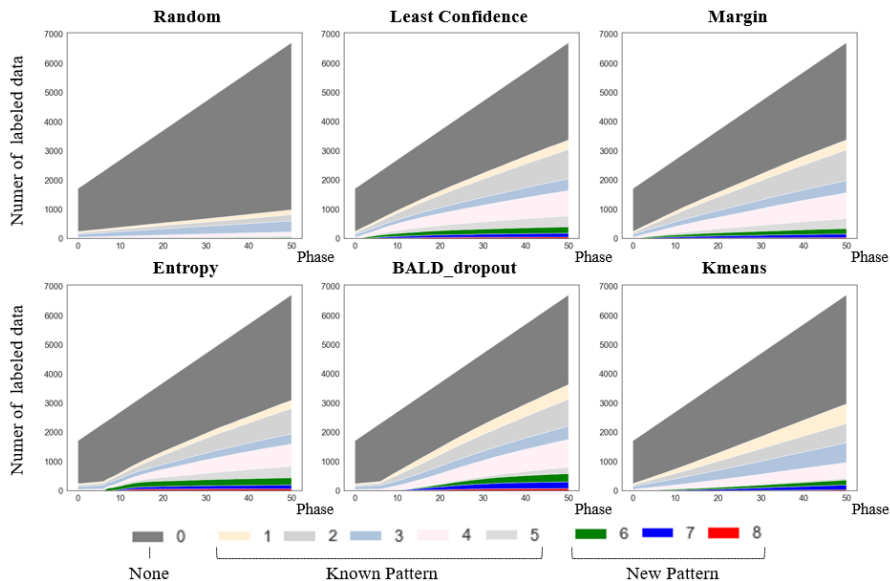


Figure 10. The Number of Labeled Training Set for Each Class as the Phases Progress.

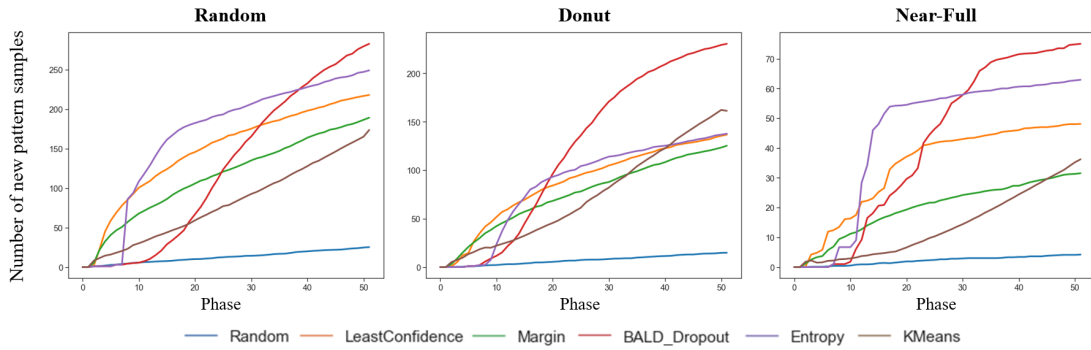


Figure 11. Number of Updated New Pattern Samples for Each Sampling Strategy

의해서는 많은 양의 신규 패턴을 확보하였다. 확률적 접근 샘플링의 경우 초기에 빠르게 검출하였으며, 이후에도 꾸준히 신규 패턴 샘플을 확보하면서 모델의 최종 분류 성능에 긍정적으로 작용하였다. 반면 베이지안 샘플링의 경우 액티브 러닝 초기에는 신규 패턴을 거의 검출하지 못했으나, phase가 진행되며 모델이 기존 패턴에 대한 어느 정도의 성능을 갖춘 이후에는 오히려 더 많은 양을 업데이트 하면서, 최종적으로는 가장 많은 수의 신규 패턴을 검출할 수 있었다. KMeans의 경우, 거리 기반 접근의 특징 때문에 신규 패턴을 우선적으로 업데이트하기보다, 지속적으로 일정한 양을 업데이트하였으며, 그 결과로 초반에는 어느 정도 양호한 성능을 보이나 최종적으로는 부족한 양이 업데이트 되었다. 결과적으로 분류 성능과 검출 성능은 유사한 곡선을 갖는 것을 확인할 수 있으며, 이는 샘플을 많이 검출할수록 모델이 학습할 수 있는 양이 늘기 때문에 분류 성능의 향상으로 이어진 것으로 보였다.

4.4 CNN 모델 변경 평가

본 장에서는 CNN 모델을 변경하여 진행했을 때, 동일하게 신규 패턴을 검출할 수 있는지 확인하고, 모델별 성능을 비교하였다. 실험은 다양한 문제에서 널리 사용되는 두가지 CNN 모델을 변경하면서 진행하였다. 첫 번째는 고전적인 CNN 모델인 LeNet5로, 앞선 실험에서 사용된 CNN 모델의 파라미터가

2,109,577개인 것에 비해 $\frac{1}{20}$ 수준인 88,193개의 파라미터를 가진 가벼운 모델이다(Lecun *et al.*, 1998). 두 번째 모델은 ResNet18이며 파라미터의 개수는 11,181,129개로, 좀 더 무겁고 깊은 딥러닝 망을 사용하였다(He *et al.*, 2016). 실험 결과는 전략마다 난수 설정을 바꾸어 가며 5회 반복하여 평균값을 사용하였으며, phase가 진행됨에 따라 증가하는 각 CNN모델의 분류 성능 결과를 <Figure 12>에 표시하였다.

실험 결과, 다른 CNN 모델을 사용해도 동일하게 신규 패턴을 초기에 검출하여 분류할 수 있었으며, 모든 모델에서 비교군인 무작위 샘플링의 부족한 성능과 대비되는 높은 성능을 보였다. 또한 모델이 변경되어도, 신규 패턴 검출에 대한 샘플링 전략의 특징은 그대로 유지되었다. 확률적 접근법이 가장 빠르게 신규 샘플을 검출하여 분류 성능을 획득하였으며, 베이지안 접근법은 초기 신규 패턴의 검출이 늦고, 거리 접근법은 초반 검출은 빠르나 후반 획득량이 적었다. 반면 모델에 따른 차이점도 존재하였다. LeNet5의 경우 모델이 가벼운 만큼 전반적으로 성능이 낮고, 초반 신규 패턴 검출이 느린 BALD_Dropout 샘플링의 경우, 오히려 무작위 샘플링보다 낮은 구간이 존재하였다. ResNet18의 경우 딥러닝 망이 깊어진 만큼, 샘플링 전략에 따른 성능 차이가 줄어들어, 모든 전략이 더 빠르게 수렴하면서 유사한 성능을 보였다.

<Table 4>에서는 각 모델의 성능에 대해 최소 신뢰 샘플링과 All_data의 F1-score로 비교하여 기술하였다. LeNet5는

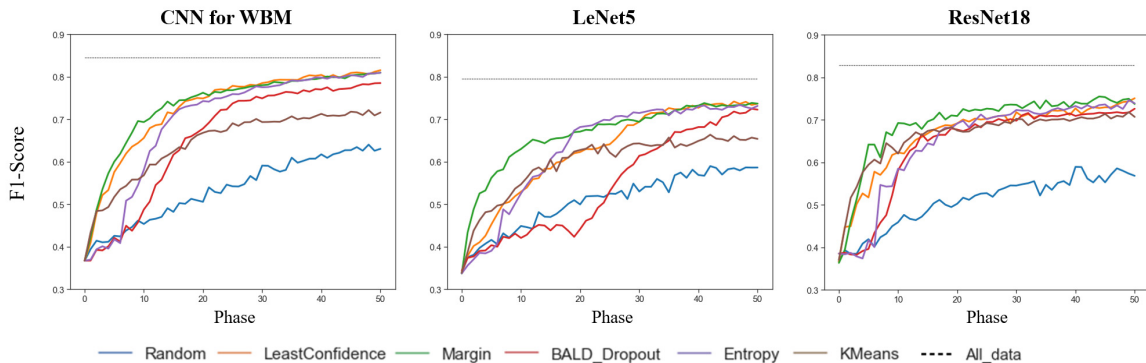


Figure 12. F1-Scores When Using LeNet5 and ResNet18

Table 4. Comparison F1-score of CNN Models

Sampling	CNN model	10 phase	20 phase	30 phase	40 phase	50 phase
Least Confidence	CNN for WBM	0.6561 (0.0587)	0.7492 (0.0313)	0.7854 (0.0084)	0.8045 (0.0108)	0.8156 (0.0076)
	LeNet5	0.5296 (0.0346)	0.6230 (0.0478)	0.6943 (0.0435)	0.7271 (0.0176)	0.7259 (0.0115)
	ResNet18	0.6232 (0.0661)	0.6917 (0.0543)	0.7181 (0.0293)	0.7272 (0.0335)	0.7510 (0.0068)
All_Data	CNN for WBM	0.8460 (0.0112)				
	LeNet5	0.7963 (0.0166)				
	ResNet18	0.8285 (0.0172)				

50phase에서 F1-score 0.7259 수준으로, 타 모델 대비 성능이 부족하였으며 ResNet18은 F1-score가 0.7510으로, 본 연구에 사용된 CNN 모델의 성능 0.8156보다 낮았다. 이는 WBM이 6464 수준의 비교적 해상도와 크기가 작은 데이터 도메인이라서, 과대적합(overfitting)되지 않도록 모델의 적정수준이 필요한 것으로 보였다. 즉, 좀더 무겁고 복잡한 이미지 데이터에서 잘 작동하는 ResNet18보다는 가볍고, LeNet5처럼 너무 가볍지 않은 수준의 CNN 모델이 최적의 성능을 보였다. 결론적으로 신규 패턴을 포함하는 액티브 러닝 구조가 다양한 모델에서 동일하게 작동하고 있음을 보였으며, 이를 이용하여 신규 클래스에 대한 다양한 문제에 대해 범용적으로 사용될 수 있을 것으로 기대된다.

5. 결론

WBM 데이터는 하루에 수 만 매 이상 생산되는 만큼, 그 안에서 신속하고 빠르게 새로운 불량을 인지하는 것이 중요하다. 아울러 공정에서는 신규 패턴이 발생하고 있는데, 그것을 인지하지 못한다면 큰 품질 사고로 이어질 수 있기 때문에 산업 현장에서는 신규 불량 탐지가 매우 중요하다. 따라서 분류 성능을 향상시키는 것뿐만 아니라 문제가 된 웨이퍼가 어떤 이력을 가졌는지 확인하기 위해서 이러한 신규 패턴 샘플을 확보하는 것 자체로 굉장히 중요한 요소가 된다. 이에 본 연구는 액티브 러닝을 통해서 신규 패턴 샘플을 확보하면서도, 실험 결과를 통해 충분히 높은 수준의 정확도를 획득할 수 있었음을 확인했다. 또한 웨이퍼 빈 맵에만 국한된 구조가 아니라 모델의 종류만 바꾼다면, 다른 모든 데이터 상황에서도 적용할 수 있기 때문에, 반도체를 포함한 다양한 산업 분야에서 기여를 할 수 있을 것으로 기대된다.

향후에는 데이터 전처리와 CNN 모델에 대한 최적화를 통해 9개 분류가 아닌 더 많은 세부 패턴으로 나누었을 때에도 높은 성능으로 작동할 수 있도록 연구하고자 한다. 본 연구는 신규

패턴 검출에 대한 구조적 관점에 집중되었기 때문에, WBM의 클래스 불균형에 의해서 발생하는 일부 클래스의 성능은 아직 모델의 고도화를 통해 개선시킬 여지가 남아있다. 아울러 샘플링 전략도 WBM 도메인에 맞추어 조정하여, 더 적은 양의 레이블링으로도 보다 빠르게 성능을 확보할 수 있는 방법론을 연구할 것이다. 나아가 최종적으로는 엔지니어가 직접 레이블링 작업을 하지 않고도 모델이 스스로 판단하여 신규 패턴을 확보하는 자동화 시스템 구축을 계획하고 있다.

참고문헌

- Alawieh, M. B., Wang, F., and Li, X. (2017), Identifying wafer-level systematic failure patterns via unsupervised learning, *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, **37**(4), 832-844.
- Chen, W. C., Tseng, S. S., and Wang, C. Y. (2005), A novel manufacturing defect detection method using association rule mining techniques, *Expert Systems with Applications*, **29**(4), 807-815.
- Gal, Y. and Ghahramani, Z. (2016), Dropout as a bayesian approximation: Representing model uncertainty in deep learning, In *International Conference on Machine Learning*, PMLR, 1050-1059.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016), Deep residual learning for image recognition, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.
- Houlsby, N., Huszar, F., Ghahramani, Z., and Lengyel, M. (2011), Bayesian Active Learning for Classification and Preference Learning, arXiv preprint arXiv, 1112.5745.
- Hsu, S. C. and 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.
- Hwa, R. (2004), Sample selection for statistical parsing, *Computational Linguistics*, **30**(3), 253-276.
- Kahng, H. and Kim, S. B. (2020), Self-Supervised representation learning for wafer bin map defect pattern classification, *IEEE Transactions on Semiconductor Manufacturing*, **34**(1), 74-86.
- Kyeong, K. and 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.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, **86**(11), 2278-2324.
- Lee, K. B., Cheon, S., and Kim, C. O. (2017), A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes, *IEEE Transactions on Semiconductor Manufacturing*, **30**(2), 135-142.
- Nakazawa, T. and 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.
- Piao, M., Jin, C. H., Lee, J. Y., and 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.
- Scheffer, T., Decomain, C., and Wrobel, S. (2001), Active hidden markov models for information extraction, In *International Symposium on Intelligent Data Analysis*, Springer, Berlin, Heidelberg, 309-318.
- Sener, O. and Savarese, S. (2017), Active learning for convolutional neural networks: A core-set approach, arXiv preprint arXiv, 1708.00489.
- Settles, B. and Craven, M. (2008), An analysis of active learning strategies for sequence labeling tasks, In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 1070-1079.
- Seung, H. S., Opper, M., and Sompolinsky, H. (1992), Query by committee, In *Proceedings of the fifth annual workshop on Computational learning theory*, 287-294.
- Shim, J., Kang, S., and 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.

- Wu, M. J., Jang, J. S. R., and Chen, J. L. (2014), Wafer map failure pattern recognition and similarity ranking for large-scale data sets, *IEEE Transactions on Semiconductor Manufacturing*, **28**(1), 1-12.
- Yoo, D. and Kweon, I. S. (2019), Learning loss for active learning, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 93-102.

저자소개

황성진 : 한양대학교 신소재공학과에서 2012년 학사학위를 취득하고 고려대학교에서 산업경영공학과 석사과정에 재학 중이다. 삼성전자 메모리사업부 연구원으로 재직 중이며, 2021년부터 고려대학교에 수학과전 중이다. 연구분야는 데이터마이닝, 머신러닝, 액티브러닝이다.

백인성 : 고려대학교 통계학과에서 2016년 학사를 취득하고, 고려대학교 산업경영공학과에서 석·박사 통합과정에 재학 중이다. 연구분야는 딥러닝, 머신러닝, Explainable AI, Game AI이다.

김성범 : 고려대학교 산업경영공학부 교수로 2009년부터 재직하고 있으며, 인공지능공학연구소 소장 및 기업산학협력센터 센터장을 맡고 있다. 미국 텍사스주립대학교 산업공학과 교수를 역임하였으며, 한양대학교 산업공학과에서 학사학위를 미국 Georgia Institute of Technology에서 산업공학 석사 및 박사학위를 취득하였다. 연구 분야는 인공지능, 머신러닝, 최적화이다.