데이터 마이닝 학회 포스터 구성

태그	
날짜	@2025년 8월 21일
상태	시작 전
실험 제목	

1. Short summary

- 초록 내용 + 전반적인 요약

Continual Learning 환경의 이상탐지는 제조업에서 새로운 제품과 결함이 순차적으로 도입되는 상황을 다루는 중요한 과제이다. 본 연구는 기존 reconstruction 방식의 한계를 극복하기 위해 Normalizing Flow를 활용하였으나, 예상과 달리 성능 저하가 관찰되었다.

분석 결과, 이는 기존 catastrophic forgetting과 다른 "spurious forgetting" 현상임을 확인하였다. Normalizing Flow의 representation은 보존되나, Gaussian Mixture Model의 클래스 중심점 간 misalignment로 인한 시적 성능 저하가 발생한다. Anomaly score를 uncertainty와 log-likelihood 기반으로 분해 분석한 결과, 전자만 최대 5배 증가하여 decision boundary 왜곡이 핵심 원인임을 확인하였다.

이러한 문제를 해결하기 위해 Gaussian Mixture Model의 구조적 정렬을 유지하면서 새로운 클래스를 점진적으로 통합하는 방법론 개발에 초점을 맞추고 있다. 본 연구는 Normalizing Flow와 Gaussian Mixture Model을 결합한 이상탐지 모델에서 spurious forgetting를 확인하고, 이를 해결하기 위한 정렬 기반 접근법을 탐구한다.

2. 배경 + 문제 정의

기존 연구 + 문제 정의

- Continual Learning 환경의 이상탐지는 제조업에서 새로운 데이터 또는 제품이 순차적으로 도입되는 상황을 다루는 중요한 과제.
- 기존에는 Reconstruction이나 Diffusion등의 방식을 기반으로 하는 방법들이 다수 제안되었으며, 이러한 방법들은 순차적으로 학습되는 class 별 특징을 식별 및 학습하기 위해 별도의 장치를 사용하곤 했음
- 대표적인 방법으로 IUF가 있음. 이 방법은 별도의 discriminator를 둠으로써 class 식별 특징을 학습하고자 했으나, 불안정하고 높은 비용의 학습, Reconstruction network의 표현 충돌 등의 문제가 있었음
- 이러한 문제를 해결하고자 기존과는 다른 방식의 architecture가 필요함을 느꼈으며, 이에 대한 대체제로 Normalizing Flow 기반의 이상 탐지를 택 함

구조의 전환 : NF

- Normalizing Flow는 가역성을 통한 강력한 지식 보존이라는 강점이 있으며, MLE를 통해 안정적으로 학습할 수 있음
- 이를 위한 베이스라인으로 HGAD를 사용. HGAD는 Multi-class AD를 위한 방법으로, 다중 클래스를 Normalizing flow와 GMM을 통해 모델링 하는 방식

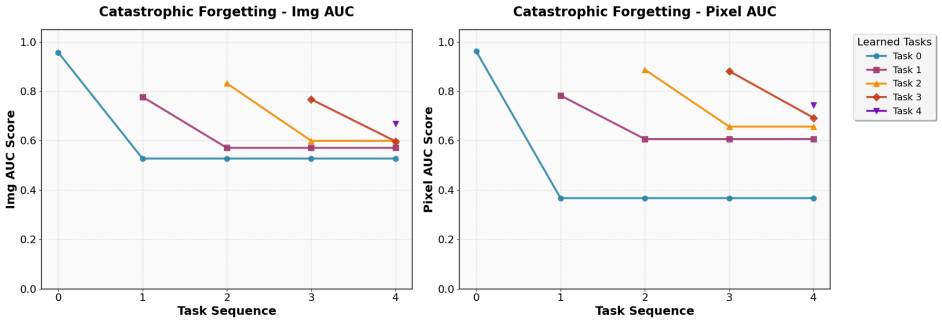
Spurious forgetting

- 실험 결과 continual learning시 성능이 저하되는 것은 NF가 knowledge를 소실했다기 보다는 GMM과의 alignment가 무너지기 때문
- 새로운 Task를 학습하는 과정에서 과거 task에 대한 NF와 GMM간의 alignment가 무너짐
- 따라서 이러한 현상을, spurious forgetting이라는 개념으로써 설명하고자 함

3. 주요 실험

1.NF into Continual Learning

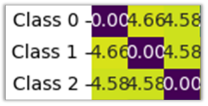
1. 성능 비교

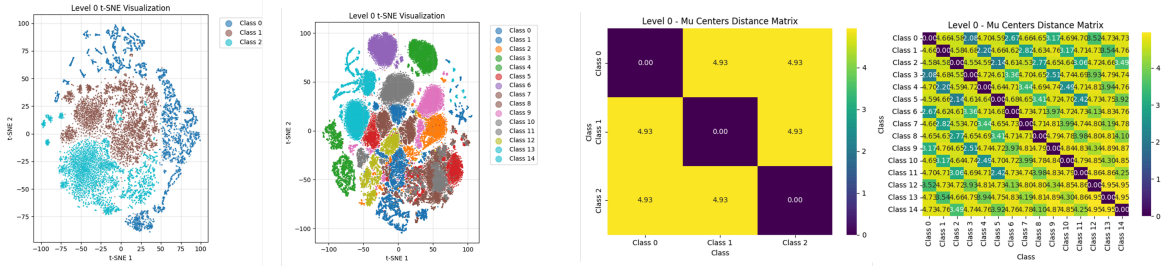


Task	Final Image AUROC	Final Pixel AUROC
0	0.44	0.34
1	0.68	0.86
2	0.59	0.83
3	0.54	0.82
4	0.97	0.93
Average	0.644	0.756

2. 임베딩 및 클래스 중심점 간 거리 시각화

a.

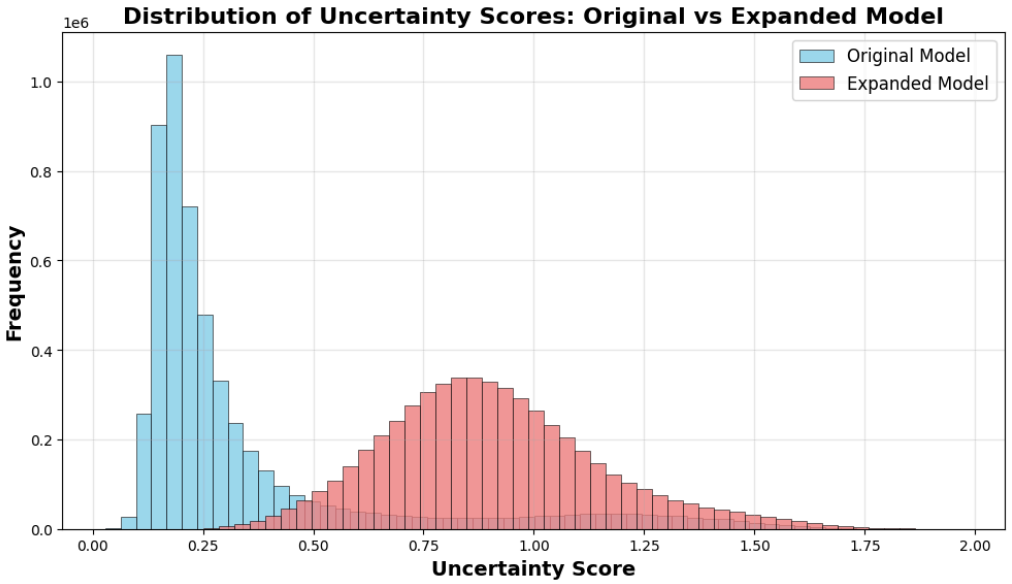




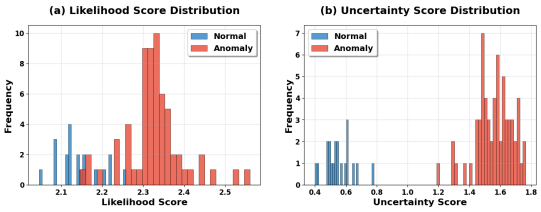
2.스코어 민감도 : Anomaly Score 분석

Task 0 학습 직후와 Task 1 학습을 위한 expand 후 비교

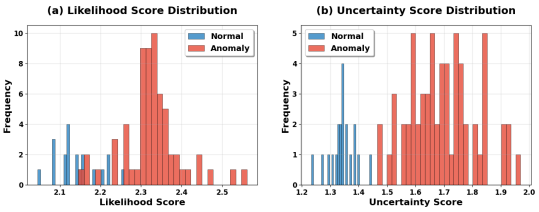
- Task 학습 직후 Anomaly score분포와, 추가 학습 없이 단순히 Task 1을 위한 GMM 요소에 대해 추가 파라미터를 추가한 경우 Anomaly score 분포를 비교 함



- 학습 직후



- expand 후

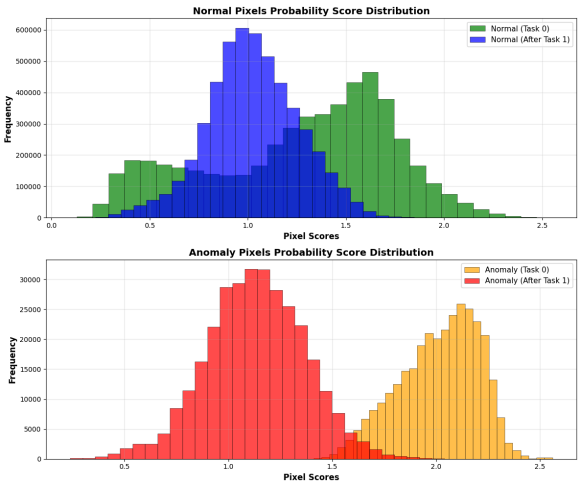


- 당연하게도 likelihood 기반의 anomaly score는 변동이 없으나, GMM의 클래스 중심점에 영향을 받는 anomaly score에는 크게 영향을 받음
- 조금 더 구체적으로 보는 경우 이상 데이터에 대한 anomaly score는 크게 변동이 없으나, 정상 데이터에 대한 Anomaly score가 전반적으로 증가하며, FPR이 증가함
  - 평균 정상 데이터의 anomaly score : 0.0869 → 0.4335 로 증가

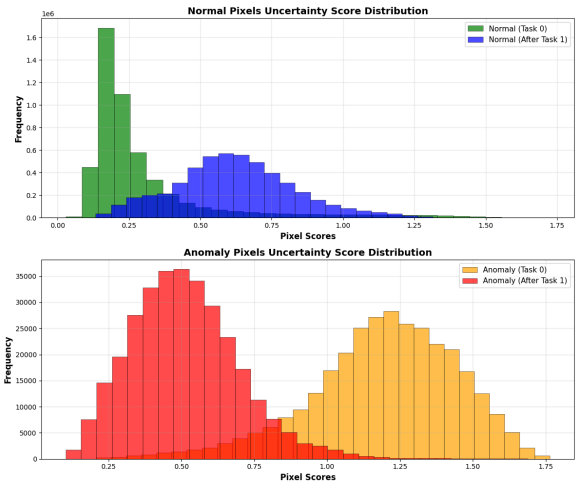
Task1 학습 후 Task0

probability score

uncertainty score



pixel-level probability score



픽셀 수준 uncertainty score

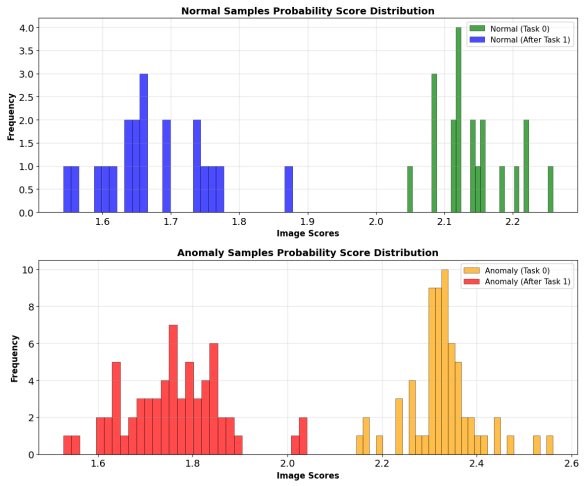
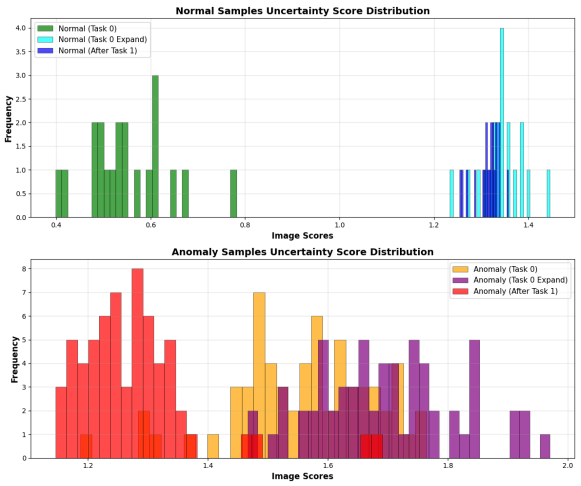


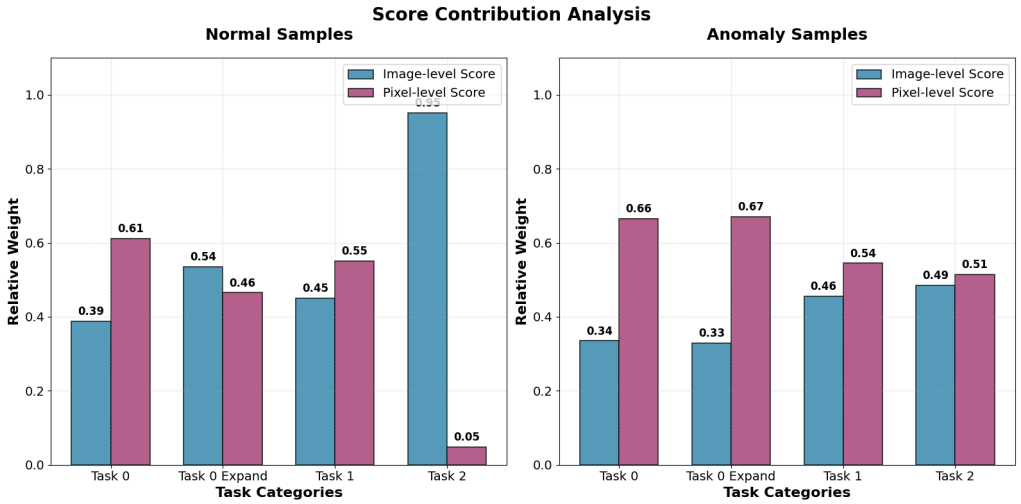
image-level probability score



이미지 레벨 uncertainty score 비교 : 이상에 대한 score는 변동이 없지만 정상에 대한 score가 계속 증가

- 최종 Anomaly score는 probability score와 uncertainty score의 곱으로 결정나기 때문에 expand 후 uncertainty score가 변함에 따라 성능이 변하게 됨
- expand 시에는 uncertainty가 크게 증가하면서 FPR이 증가했던 것 과 달리, Task 1 학습 후에는 Normal과 Anomaly에 대한 score가 전반적으로 모두 낮아짐. 이에 따라 이상을 정상으로 식별하는 False Negative 가 증가

최종 score 기여도 분석



- probability 기반의 score와 uncertainty score가 최종 anomaly score에 기여하는 가중치를 task 진행에 따라 비교한 결과
- Task 2 학습 이후 Task 0의 score에 대해 uncertainty의 가중치가 score의 가중치
- **정상 샘플(왼쪽)**
  - **Task0→Expand→Task1**에서 uncertainty 비중이 다시 커졌다가(Task1에 0.55), **Task2에서 급감(0.05)** 하고 probability가 거의 전부(0.95).
  - 해석: **Task1 전환 시에는 클래스 중심 정렬이 잠깐 흐트러져**(posterior가 퍼져서 H(r)H(r)H(r)↑), 정상조차 불확실성의 영향을 크게 받음. 이후 **Task2까지 가며 중심 재정렬**이 되면서 posterior가 뾰족해지고 H(r)H(r)H(r)↓ → **likelihood(확률) 신호가 지배**.
  - 결론: 정상에 대해서는 **"정렬 안정화 → likelihood 주도"**로 수렴하는 모습.

- **이상 샘플(오른쪽)**
  - 전 시점에 걸쳐 **uncertainty가 여전히 우세(0.67→0.54→0.51)**. probability 기여도는 점진적 상승 (0.33→0.46→0.49).
  - 해석: 이상은 **어느 중심에도 확실히 속하지 않는 상태**가 유지(엔트로피가 높음). 학습이 진행되며 경계가 선명해져 **likelihood도 일부 분리 신호**를 보태지만, **탐지는 여전히 uncertainty-driven**.
- **전환 시점 신호**
  - Expand/Task1에서 정상의 uncertainty 비중이 커진 건 **정렬 붕괴(centers drift)**의 정량적 징후. 이후 다시 probability가 커진 건 **사후 재정렬/적응**이 이루어졌다는 증거.

3. Task Misalignment 확인 → spurious forgetting (misalignment)가 문제임을 확인

1. 정렬-원인 규명 실험 (파라미터 고정, 분해 + concept drift)

- **공통 패턴:** 네 조건 모두에서 ΔH_normal<0 → 정상 샘플은 항상 “더 확신 있게” 분류됨. 결과적으로 FPR95는 모든 조건에서 감소.
- **차이점:**
  - **C (둘 다 update):** 정상/이상 분리 가장 잘 됨 → 성능 개선 최대.
  - **B (NF update만):** 이상 엔트로피 ↑, 정상 엔트로피 ↓ → “forward adaptation” 효과.
  - **A (GMM update만):** 이상도 entropy ↓ → 이는 분리보다는 **정렬 자체가 흔들리면서 전체 확신이 높아진 착시** 가능성.
  - **D (둘 다 freeze):** baseline 변화만 보임 → drift 없이도 소폭 수치변화.

> ΔH/ΔFPR 분석 결과, 네 조건 모두 정상 샘플의 엔트로피는 감소(확신 증가)하고 FPR95는 감소했으나, 이상 샘플의 엔트로피 변화 패턴은 조건마다 달랐다. NF와 GMM을 함께 업데이트(C) 하면 정상은 확신↑, 이상은 불확실성↑가 동시에 일어나며 FPR이 가장 크게 감소하였다. 반면 GMM만 업데이트(A) 하면 정상·이상 모두 entropy가 줄어 posterior가 날카로워졌으나 이는 정렬 붕괴의 부산물일 수 있다. NF만 업데이트(B) 한 경우에는 이상 엔트로피가 증가하고 정상은 감소하여 forward adaptation 효과가 나타났다. Freeze(D) 조건에서는 baseline 수준의 소폭 변화만 있었다.
>
> ⇒ 즉, 성능 저하/개선의 주 요인은 **정렬 붕괴(GMM 이동)**와 **NF 표현 적응**의 상호작용이며, 엔트로피/FPR 지표를 통해 이를 수치적으로 구분할 수 있었다.

4. Post-hoc 소량 정렬 (Few-shot)

	Image AUROC			**Pixel AUROC**		
	Bottle	**Cable**	**Capsule**	**Bottle**	**Cable**	**Capsule**
Task 1 학습 후	1	0.946	0.979	0.985	0.859	0.99
Task 2 학습 후	**0.654**	**0.519**	**0.491**	**0.927**	**0.586**	**0.657**
Few-shot training	**0.998**	**0.719**	**0.837**	**0.956**	**0.722**	**0.950**

- Task 2 학습 후 Task 1의 성능은 매우 낮게 나타남
- 그러나 Task 1의 데이터로 약간의 학습을 해주게 되면 바로 학습 직후에 준하는 성능으로 복구 됨
- 실제 지식이 forgetting이 된 것이 아닌, GMM의 정렬이 잘못되어 성능이 저하되는 것으로 추측
- 일종의 spurious forgetting[5]이라고 볼 수 있음
- 실제 지식이 소실된 것이 아니라, 과거 task에 대한 정렬이 손실되었기 때문에 성능이 저하되는 것

-

5. 기존 방법론 적용

1. 기존 방법론을 통해 misalignment 문제를 해결할 수 있는지를 확인

	Image-AUROC	Pixel-AUROC	Image-Forgetting	Pixel-Forgetting
HGAD(SFT)	0.646	0.753	-0.334	-0.207
+EWC	0.520	0.797	-0.167	-0.066
+LWF	0.646	0.750	-0.333	-0.209
+PackNet	0.647	0.757	-0.332	-0.206
+Replay	**0.952**	**0.975**	**-0.030**	**-0.002**

- 대부분의 방법은 NF 자체에 Regularization을 가하는 방법이기 때문에 성능 차이가 크게 발생하지 않음
- 그러나 Replay의 경우 과거 Task의 데이터를 저장한 뒤 다시 학습에 사용하는 방법이기 때문에 GMM의 클래스 중심에 영향을 줌
- 따라서 최종 결과가 multi-class에 준하는 성능이 나옴

Findings?

- Forward transfer 능력
- Task=0 첫 Task 학습할 때는 수렴까지 오랜 시간이 소요되지만, 그 이후 Task 부터는 굉장히 빠른 속도로 수렴 됨