설비의 알람 유형 분류를 위한 오픈셋 인식 심층 신경망

Open Set Recognition Deep Neural Network for Classification of Facility Alarm Types

김상훈·김성범

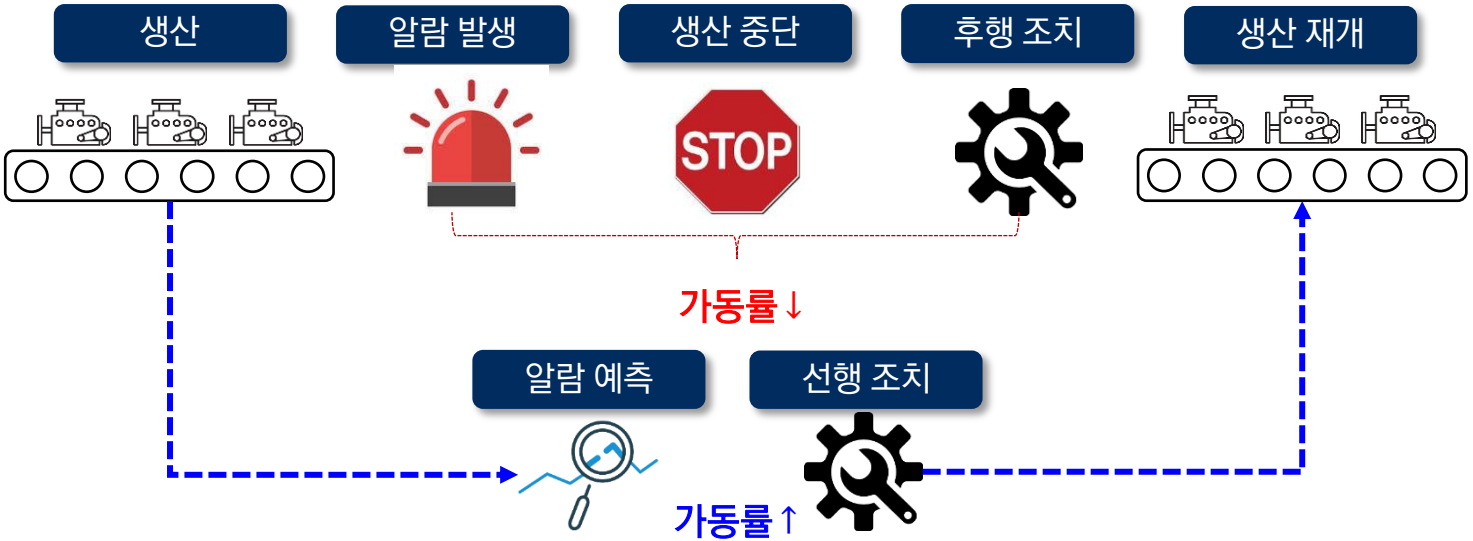
고려대학교 산업경영공학과

목차

연구 개요

연구 배경

- 하드웨어와 소프트웨어의 발전에 따라 공정 진행 중 알람 프로세스 **실시간 모니터링**이 가능
- 공정의 전반적인 복잡성을 증가에 따라 알람 시스템 모니터링 및 프로세스 운영에 높은 수준의 대응이 요구
- 공정 문제를 야기하는 **중요 알람을 조기 예측**을 통한 선행적 설비 점검 → 안전과 비용적 측면에서 매우 효율적



연구 개요

심층 신경망을 활용한 알람 조기 예측

- N-gram 모델을 사용하여 각 유형별 알람 발생 확률을 예측 (Zhu et al., 2016) → 장기적인 시계열 패턴을 고려하지 못함.
- RNN 기반의 LSTM 모델을 사용하여 장기적 시계열 패턴을 반영한 알람 유형 분류 (Cai et al., 2019)
- 채널 별 특징 추출을 위해 1차원 CNN 모델을 이용하여 알람 유형 분류 (Ince et al., 2016)

- 기존 심층 신경망 기반 방법론은 모든 예측 가능한 클래스가 학습된다는 **닫힌 세계 가정**을 따름(Fei & Liu, 2016).
  - 실제 공정에서 **정의된 알람 유형은 수백 가지**에 이름
  - 빈번하게 발생하는 알람 유형이 있는 반면 실제로 **거의 발생하지 않는 알람 유형이 존재**
  - 데이터 **수집 기간 안에 모든 클래스의 알람 데이터를 수집하는 것은 불가능**
- 학습하지 않은 클래스의 데이터에 대해 예측할 때 높은 확률로 학습된 클래스 중 하나로 **오분류**

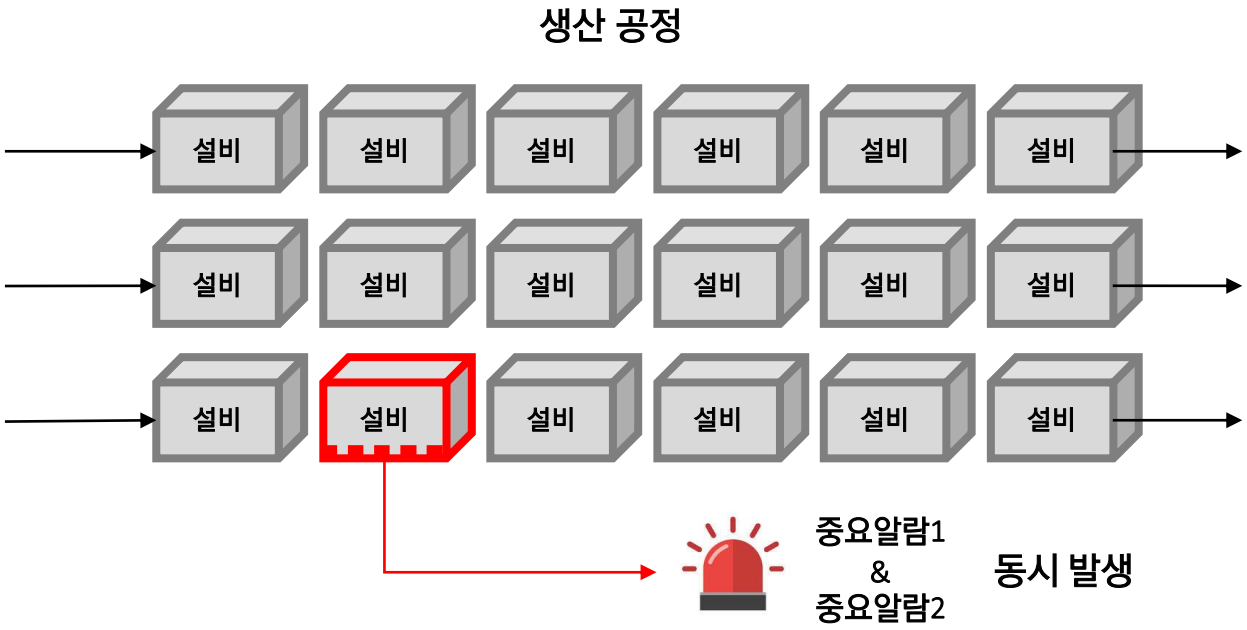
➡ 학습하지 않은 클래스에 대하여 모르는 클래스로 분류하여 거부하는 **오픈셋 분류 방법론** 필요

1. Zhu, J., Wang, C., Li, C., Gao, X., & Zhao, J. (2016). Dynamic alarm prediction for critical alarms using a probabilistic model. Chinese Journal of Chemical Engineering, 24(7), 881-885.
2. Cai, S., Palazoglu, A., Zhang, L., & Hu, J. (2019). Process alarm prediction using deep learning and word embedding methods. ISA transactions, 85, 274-283.
3. Ince, T., Kiranyaz, S., Eren, L., Askar, M., & Gabbouj, M. (2016). Real-time motor fault detection by 1-D convolutional neural networks. IEEE Transactions on Industrial Electronics, 63(11), 7067-7075.
4. Fei, G., & Liu, B. (2016, June). Breaking the closed world assumption in text classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational

연구 개요

다중 레이블 분류 상황

- 실제 공정에서 알람은 **동시 다발적으로 발생**
- 다중 레이블 분류 모델 구축 필요



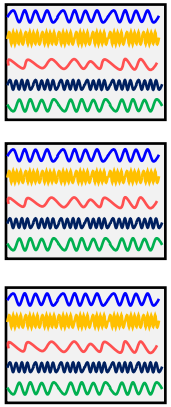
연구 개요

다중 레이블 분류 상황

- 실제 공정에서 알람은 동시 다발적으로 발생
- **다중 레이블 분류 모델** 구축 필요

다중 레이블 분류모델: Y=F(X)

입력 데이터(X)



정답 (Y)

정상	알람1	알람2	알람3	알람4	알람5	알람6
1	0	0	0	0	0	0
0	1	0	0	1	0	0
…	…	…	…	…	…	…
…	…	…	…	…	…	…
0	1	0	0	0	1	0
0	1	1	0	0	0	0

관련 연구

오픈셋 분류 모델 : (1) 초기의 오픈셋 분류

- 초기의 오픈셋 분류는 **각 클래스별로 독립적인 이진 분류 모델**을 구축하는 방식으로 이루어 짐
- **모든 클래스에서 거부될 경우** 학습되지 않은 유형으로 분류
- 각 클래스별로 One-class SVM를 구축하여 학습하지 않은 클래스 거부 (Sch¨olkopf et al., 2001)
- 각 클래스의 중심에 대한 유사도 벡터를 활용하는 CBS (center-based similarity) 공간 학습 방법을 통해 클래스별 이진 분류기를 구축 (Fei & Liu, 2015)
- 클래스별 이진 SVM 분류기에서 분류 결정 경계면을 구축할 때 학습된 클래스의 데이터가 거의 없는 공간(open set risk)을 배제하여 학습하지 않은 클래스 거부 성능 향상 (Scheirer et al., 2012)

1. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. Neural computation, 13(7), 1443-1471.Cai, S., Palazoglu, A., Zhang, L., & Hu, J. (2019). Process alarm prediction using deep learning and word embedding methods. ISA transactions, 85, 274-283.
2. Fei, G., & Liu, B. (2015, September). Social media text classification under negative covariate shift. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 2347-2356).
3. Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., & Boult, T. E. (2012). Toward open set recognition. IEEE transactions on pattern analysis and machine intelligence, 35(7), 1757-1772.

관련 연구

오픈셋 분류 모델 : (2) 심층 신경망에 오픈셋 분류 적용

- 기존 심층 신경망을 활용한 분류기의 소프트 맥스 함수를 극단치 이론에 기반하여 수정 → 학습하지 않은 클래스의 소프트 맥스 확률값 도출 (Bendale & Boult, 2016)
- CNN 특징 추출기에 시그모이드 출력 레이어를 적용한 다중 레이블 분류 모델을 차용하여 단일 레이블 오픈셋 분류를 수행 (Shu et al., 2017)

기존 방법론

- 학습 클래스와 같은 도메인의 학습하지 않은 클래스에 대한 **오픈셋 분류 성능 수준이 낮음**
- **하이퍼 파라미터**에 따라 결과가 **민감**하게 반응
- 주로 **단일 레이블 분류 상황**에서 오픈셋 분류 연구가 이루어 짐

제안 방법론

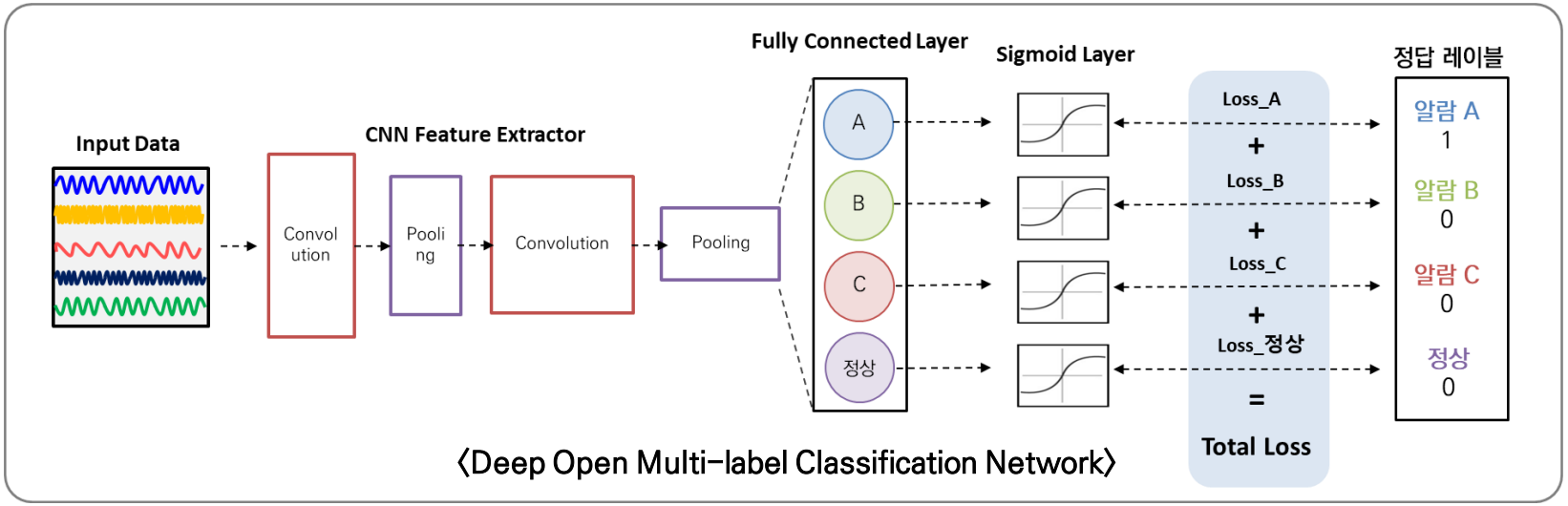
1. 배경 데이터를 활용한 방법론을 제안하여 학습 클래스와 **같은 도메인의 학습하지 않은 클래스에 대해서** 오픈셋 분류 성능이 기존의 방법론보다 **우수함을 보임**
2. 실제 공정에서 동시다발적으로 발생한 알람 유형 데이터를 사용하여 **다중 레이블 분류 상황에서 오픈셋 분류 알고리즘**을 적용함

1. Bendale, A., & Boult, T. E. (2016). Towards open set deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1563-1572).
2. Shu, L., Xu, H., & Liu, B. (2017). Doc: Deep open classification of text documents. arXiv preprint arXiv:1709.08716.

제안 방법론

모델 구조 (1) Feature Extractor

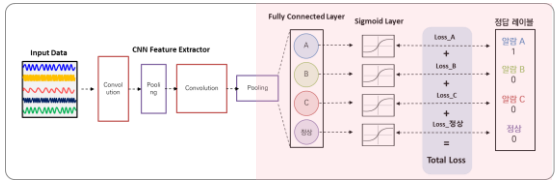
- 특징 추출기와 시그모이드 출력 레이어를 결합한 다중 레이블 분류 모델
- 특징 추출기는 RNN 및 LSTM 모델로 대체 가능
- 대부분의 오픈셋 분류 알고리즘이 **CNN 특징 추출기** 기반이므로 공정한 비교를 위해 CNN 특징 추출기 채택
- 다채널 시계열 데이터의 특징 추출에 좋은 성능을 보이는 **1차원 convolution 연산을 사용**하는 CNN 특징 추출기를 활용



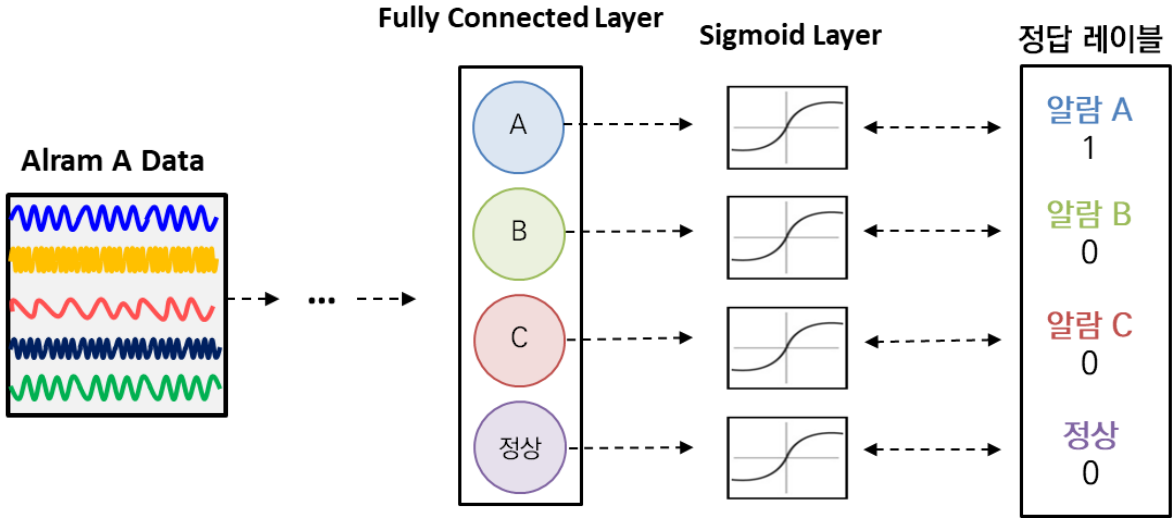
〈Deep Open Multi-label Classification Network〉

제안 방법론

모델 구조 (2) Sigmoid Layer

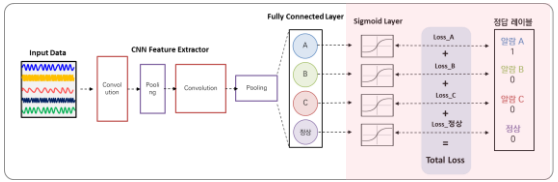


1. 특징 추출기로 추출한 채널별 특징 벡터를 Fully Connected Layer에 연결하여 학습 클래스 수와 동일한 차원의 벡터로 크기를 줄임
2. 축소된 벡터를 학습 클래스의 수만큼 시그모이드 함수를 포함하는 시그모이드 출력 레이어에 연결
3. 각 i 번째 시그모이드 함수는 학습데이터에서 클래스(c_i)에 해당하는 i 번째 레이블의 정답이 1 인 데이터를 **정답 학습 데이터**로, 정답이 0인 나머지 모든 데이터를 해당 클래스에 **오답 학습 데이터**로 사용



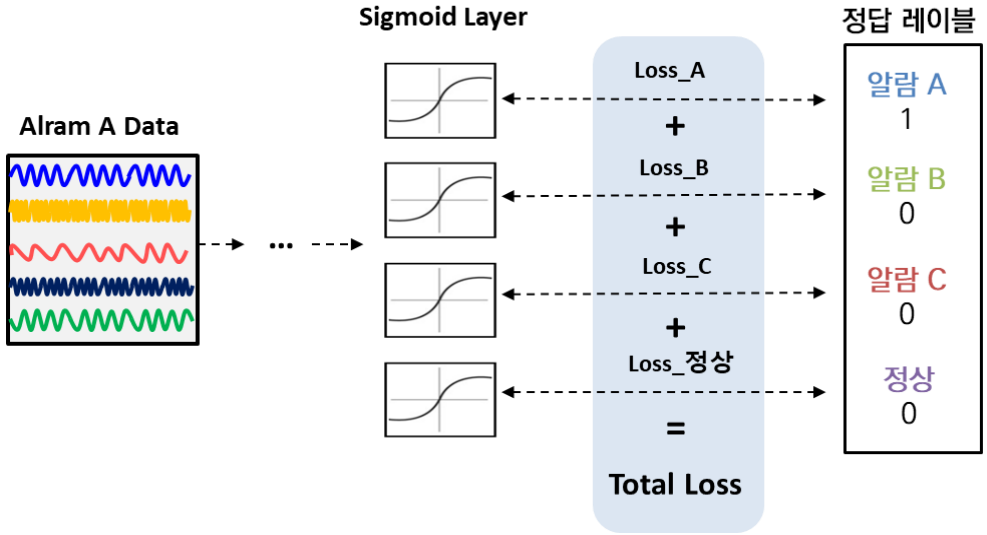
제안 방법론

모델 구조 (2) Sigmoid Layer



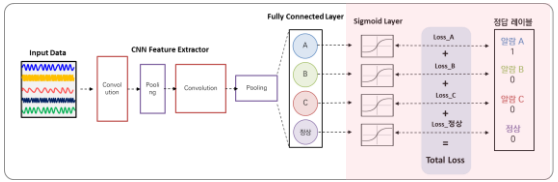
- 학습 데이터에 대한 모든 클래스별 시그모이드 함수가 출력하는 확률 값의 **로그 손실 총합**을 손실함수로 두어 모델을 학습
- **손실함수**

$$Loss = -\sum_{i=1}^{C}\{y_i log\,g(f(s_i)) + (1-y_i)log(1-f(s_i))\}$$



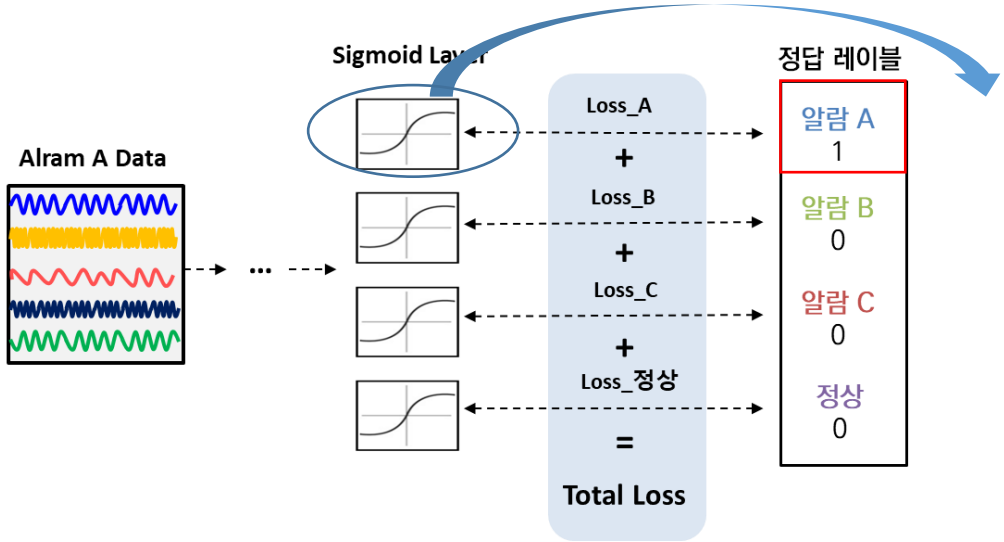
제안 방법론

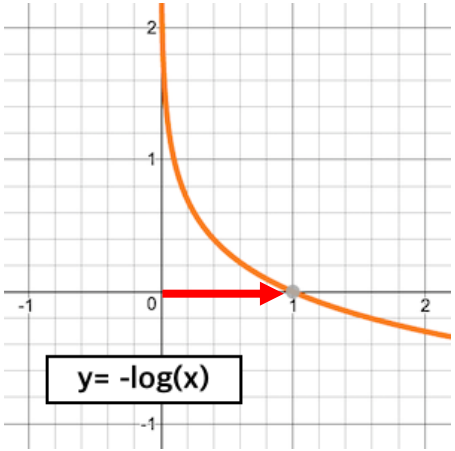
모델 구조 (2) Sigmoid Layer



- **정답 학습 데이터** : 손실함수에 따라 정답 레이블이 1인 클래스의 시그모이드 출력 확률은 **1에 수렴**

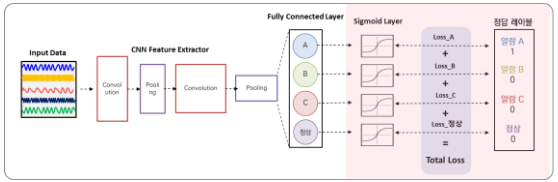
$$Loss\_A = -log\left(f(s_i)\right)$$





제안 방법론

모델 구조 (2) Sigmoid Layer

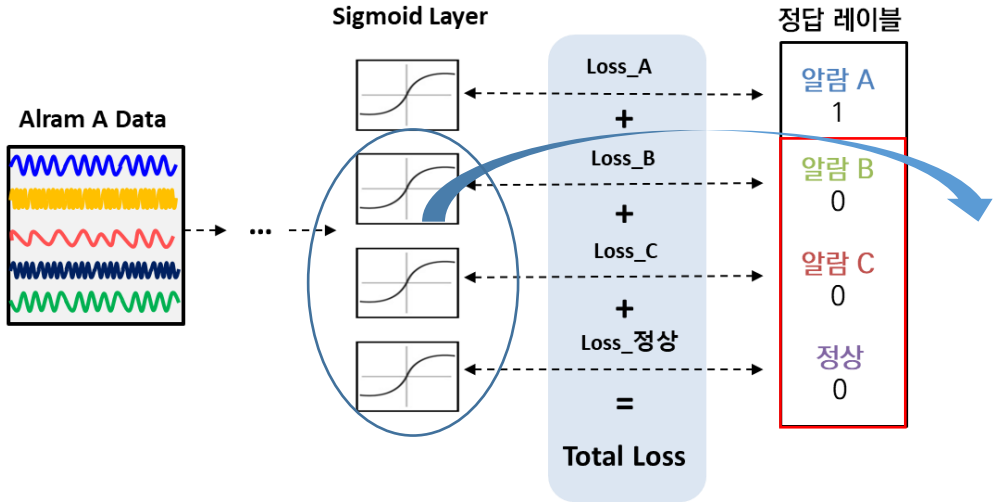


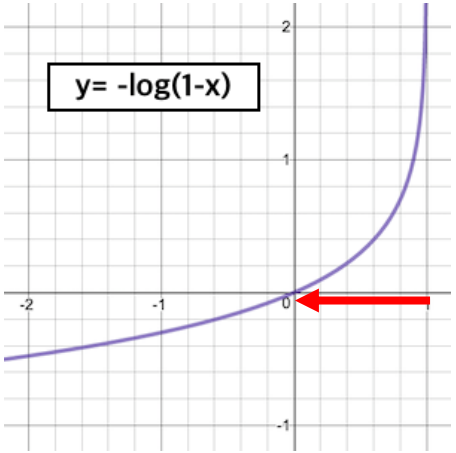
- **오답 학습 데이터** : 손실함수에 따라 정답 레이블이 0인 클래스의 시그모이드 출력 확률은 **0에 수렴**

$$Loss_B = -log\left(1 - f(s_i)\right)$$

$$Loss_B = -log\left(1 - f(s_i)\right)$$

$$Loss_{normal} = -log\left(1 * f(s_i)\right)$$



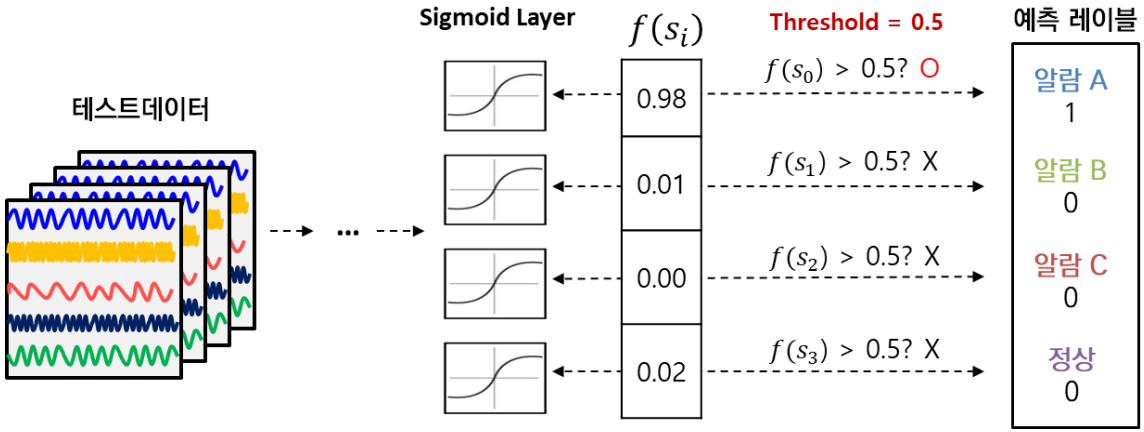


제안 방법론

모델 구조 (3) Threshold

- 모델 학습이 완료되면 새로운 데이터 x_i가 입력되었을 때 모델은 시그모이드 출력 확률 $f(s_i)$를 산출
- 각 클래스별 시그모이드 출력 확률의 **임계 값(t_i)**보다 크면 i번째 클래스 예측 레이블은 1이 되고 반대로 임계 값보다 작다면 i번째 클래스 예측 레이블은 0이 됨
- 일반적으로 각 클래스별 시그모이드 출력 확률의 임계 값은 0.5를 사용

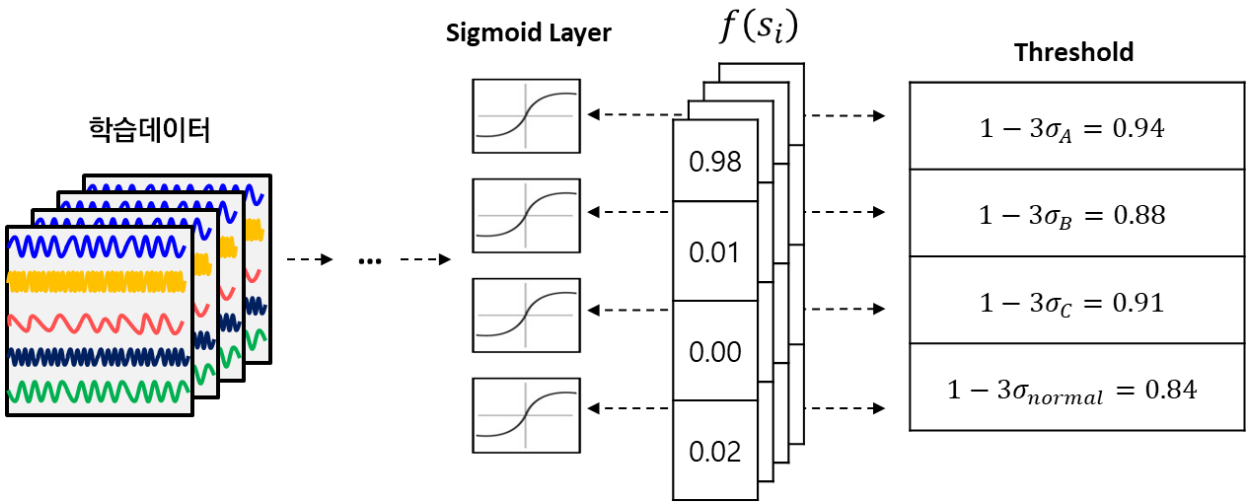
$$\hat{y}_i = \begin{cases} \hat{y}_i = 1, & if\ f(s_i) > t_i \\ \hat{y}_i = 0, & if\ f(s_i) < t_i \end{cases}$$



제안 방법론

모델 구조 (3) Threshold

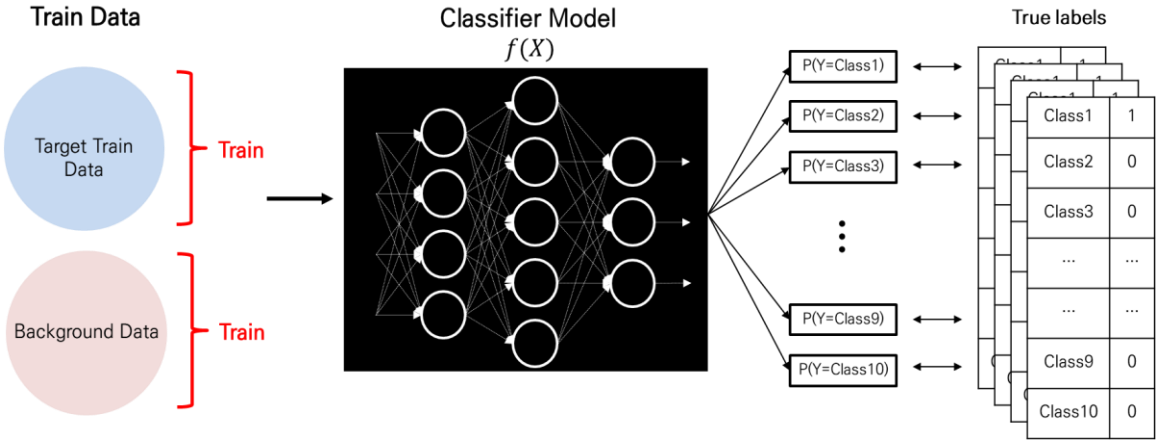
- 손실 함수에 따라 모델이 학습되면 정답 학습 데이터에 대한 시그모이드 출력 확률이 1에 가까운 값에 수렴
- 임계 값을 0.5로 사용할 경우 **분류 결정 경계가** 학습 클래스의 데이터가 거의 없는 **열린 공간을 배제하지 못함**
- **정답 학습 데이터의 클래스별 시그모이드 출력 확률의 분포**를 반영하여 임계 값을 설정하면 오픈셋 분류 성능 향상
- 클래스별 시그모이드 출력 확률이 **평균이 1인 가우시안 분포의 절반**을 따른다고 가정
- i번째 클래스의 임계 값을 **가우시안 분포의 표준 편차(σ_i)**를 사용하여 $1-\alpha\sigma_i$로 대체 (α는 하이퍼파라미터, 대개 3을 사용)



제안 방법론

Background Data 사용

- 정답 학습 데이터를 기반으로 임계 값을 수정하는 방법론은 **오답 학습 데이터를 분류 결정 경계면에 반영시키지 못함**
- 데이터 분포에서 멀리 떨어진 어떠한 학습 클래스에도 속하지 않는 데이터에 대해서 **높은 확률로 모든 학습 클래스에서 아니라고 거부**할 수 있어야 함
- 모델의 학습 단계에서 학습 클래스에 속하지 않는 **배경데이터(background data)를 활용**하면 학습 클래스가 아닌 데이터에 대한 거부 확률을 높일 수 있음



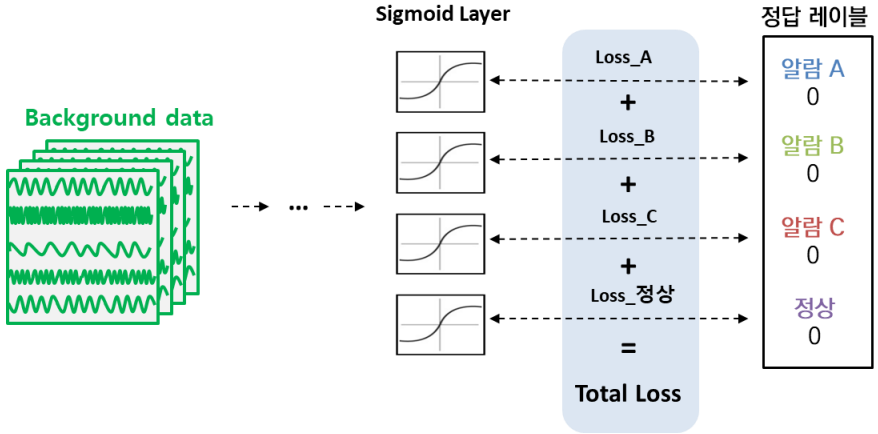
제안 방법론

Background Data 사용

- 배경데이터(background data)를 활용하는 오픈셋 분류 알고리즘의 경우 배경데이터에 대한 **손실함수의 정의가 필요**
- 배경데이터에 대해 새롭게 정의한 손실함수를 기존의 손실함수에 더하여 전체 손실함수를 재정의

$$Loss_{Total} = Loss_{Train} + Loss_{Background}$$

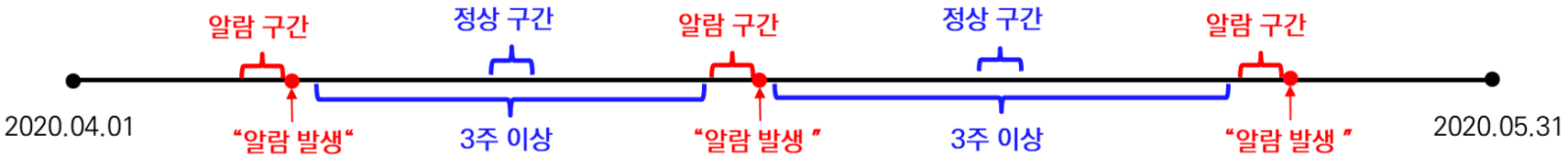
- 손실함수를 각각 정의하여 더할 경우 모델의 학습과정에서 **상대적으로 쉬운 작업에 해당하는 손실함수가 빠르게 수렴**하는 한계
- 제안 모델 구조에서는 각 클래스별 정답 레이블에 따라 학습이 진행하므로 **손실함수를 새롭게 정의하지 않고** 배경데이터의 모든 **클래스 레이블 정답을 0으로 입력**하여 학습하면 모든 클래스에서 거부되는 예로 학습에 반영됨



실험 결과

(1) 데이터 소개

- 자동차 제조 업체 A사의 엔진 공장의 헤드 가공 공정 총 12개의 설비에서 2개월간 수집된 42채널의 설비 상태 데이터
    1. 학습 클래스 데이터 : **정상 유형**과 헤드 가공 공정의 **중요 알람 유형 6종**
    2. 오픈셋 분류 성능 평가 데이터 : 헤드 가공 공정의 **중요하지 않은 알람 유형 3종**
    3. 배경 데이터 : 다른 공정에서 수집된 42채널 설비 상태 데이터
- **알람 수집 시점 3일 전까지의 기간**을 해당 알람 구간으로 정의
- 정상 구간은 최대한 알람의 전조 증상을 배제하기 위해 알람 발생 간격이 3주 이상인 구간에서 일부 추출



실험 결과

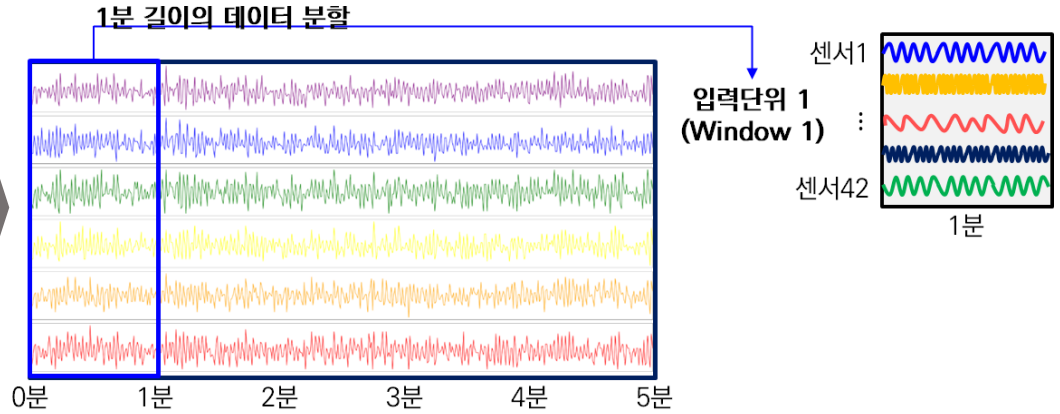
(2) 데이터 전처리

- 데이터 수집 간격을 1초 간격의 평균으로 요약하여 **시간 단위를 통일**
- 모델에 입력될 한 관측치 단위를 **60초 간격의 윈도우로 설정**

데이터 수집 간격 통일(평균)

날짜 및 시간	Value
2020-01-21 00:00:01.0	0.2
2020-01-21 00:00:01.2	0.3
2020-01-21 00:00:01.8	0.1
2020-01-21 00:00:02.1	0.1
2020-01-21 00:00:02.3	0.15
2020-01-21 00:00:02.4	0.2

날짜 및 시간	Value
2020-01-21 00:00:01	0.2
2020-01-21 00:00:02	0.15
2020-01-21 00:00:03	0.1
2020-01-21 00:35:30	0.1
2020-01-21 00:35:31	0.1
2020-01-21 00:35:32	0.2



모든 데이터에 동일한 전처리 과정 수행

실험 결과

(2) 데이터 전처리

- 전체 데이터 입력단위를 학습 데이터 6,745개, 테스트 데이터1,413개로 분리
- 배경데이터는 385개로 모든 클래스의 레이블 값이 0인 형태로 구성하여 학습 단계에 포함
- 모델 학습에 참여하지 않은 데이터(unseen data)를 2,995개로 구성하여 오픈셋 분류 성능 평가에 사용

Class	Training data	Testing data	Total
Normal	1,818	375	2,193
Alarm 1	1,153	234	1,387
Alarm 2	1,437	302	1,739
Alarm 3	267	55	322
Alarm 4	646	124	770
Alarm 5	441	85	526
Alarm 6	787	189	976
Alarm 1 + Alarm 2	64	16	80
Alarm 1 + Alarm 5	132	33	165
Total (Seen)	**6,745**	**1,413**	**8,158**
Background	385	-	385
Unseen	-	2,995	2,995
Total (Unseen)	**385**	**2,995**	**3,380**

실험 결과

(3) 실험 설계

1. 오픈셋 알고리즘을 반영하지 않은 다중 레이블 분류 모델 학습 후 성능 평가
    - 알람 유형별 공정 센서 데이터에 오픈셋 알고리즘 적용 효과 검증

2. 최신 오픈셋 분류 알고리즘 기법인 DOC 기반의 다중 레이블 오픈셋 분류 모델 (Shu et al., 2017)과 극단치 이론 기반의 다중 레이블 오픈셋 분류 모델 (Jang & Kim, 2020)과 성능 비교
    - 제안 방법론의 오픈셋 분류 성능 우수성 검증

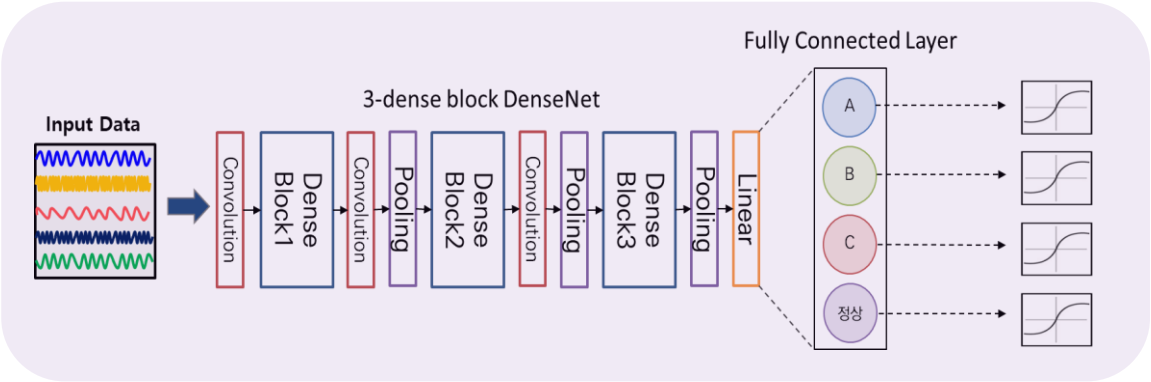
3. 배경데이터를 모든 학습 클래스에 대해 거부되는 예가 아닌 하나의 새로운 학습 클래스로 추가하여 학습한 모델과 비교
    - **배경데이터의 활용 방식**의 효과 검증

1. Shu, L., Xu, H., & Liu, B. (2017). Doc: Deep open classification of text documents. arXiv preprint arXiv:1709.08716.
2. Jang, J., & Kim, C. O. (2020). One-vs-Rest Network-based Deep Probability Model for Open Set Recognition. arXiv preprint arXiv:2004.08067.

실험 결과

(3) 실험 설계

- 비교 방법론들은 모델 학습이 완료된 후 **후처리를 통해 오픈셋 분류를 하는 방법론**
- 모델 구조에 제약이 없어 마지막 레이어를 제외하곤 모델의 구조를 자유롭게 변경 가능 (특징 추출기)
- 모든 비교 방법론에서 3-dense block DenseNet에 시그모이드 출력 레이어를 결합한 모델 구조를 적용
- 학습 데이터를 훈련:검증 = 8:2로 분할하여 모델 학습 진행
- 모든 방법론에서 훈련, 검증 데이터의 분할 시드를 변경해가며 총 20회 반복실험



모델 하이퍼파라미터 세팅

- ▶ Batch size : 128
- ▶ Optimizer : Adam
- ▶ Epoch : 100
- ▶ Initial Learning rate : 0.001
- ▶ Learning rate decay rate after 10 epoch : 0.9
- ▶ Early Stopping Patience : 15

실험 결과

(4) 평가 지표

- 다중 레이블 분류 성능 지표로 많이 사용 되는 Jaccard similarity score 지표를 통해 성능 비교
- 각 클래스별 이진 분류 수행 시 실제 거짓이 실제 참값에 비해 많으므로 정밀한 비교를 위해 F1 score로 클래스별 분류 정확도를 평가

$$Jaccard\ Similarity\ Score = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|},$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1\text{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall_i}$$

Class K		Predicted Class	
		C_K	$Not\ C_K$
Actual Class	C_K	TP	FP
	$Not\ C_K$	FN	TN

[클래스별 Confusion Matrix]

실험 결과

(5) 실험 결과

- 학습 클래스에 대한 분류 성능은 모든 방법론에서 비슷하게 나타남
- **학습하지 않은 클래스**에 대한 감지 성능은 **제안 모델에서 가장 우수한 성능**을 보임
- 배경데이터를 모든 클래스에 대해 오답인 예제로 학습에 추가하면 학습 클래스 분류 성능을 유지하면서 학습하지 않은 클래스의 분류 성능이 높게 나오는 것을 확인

Algorithm	Jaccard Similarity Score			F_1 Score		
	Seen classes	Unseen classes	Overall classes	Seen classes	Unseen classes	Overall classes
MLC(t=0.5)	**0.9767 (0.0221)**	0.2805 (0.0285)	0.4996 (0.0264)	**0.9782 (0.0224)**	0.2805 (0.0225)	0.4998 (0.0225)
DOC	0.9727 (0.0173)	0.6357 (0.0244)	0.7530 (0.0198)	0. 9733 (0.0154)	0.6357 (0.0214)	0.7532 (0.0175)
EVT-based MLC	0.9727 (0.0208)	0.3419 (0.0533)	0.5142 (0.0415)	0.9766 (0.0242)	0.3219 (0.0432)	0.5108 (0.0391)
Extended DOC	0.9566 (0.0196)	0.5830 (0.0452)	0.7209 (0.0299)	0.9572 (0.0254)	0.5830 (0.0411)	0.7211 (0.0301)
Proposed method (background)	0.9729 (0.0114)	**0.8407 (0.0268)**	**0.8789 (0.0187)**	0.9730 (0.0217)	**0.8417 (0.0325)**	**0.8791 (0.0298)**

[20회 반복실험에 따른 분류성능표]

실험 결과

(5) 실험 결과

- 제안 방법론은 다른 방법론에 비해 각 클래스별 **정답 학습 데이터의 시그모이드 출력 확률 값이 편차가 작으면서 높게 나타남**
  - 다른 방법론에 비해 높은 기준의 분류 경계 임계 값이 생성되어 더욱 엄격하게 해당 클래스에 속하는지를 판정
- 오답 데이터 기준 시그모이드 출력 확률 값이 학습, 테스트 모두에서 가장 낮게 나타남
  - 학습하지 않은 클래스의 데이터에 대하여 더욱 확실하게 오답으로 예측

Algorithm	Training data		Testing data	
	Positive	Negative	Positive	Negative
DOC	0.9987 (0.0264)	0.0002 (0.0027)	**0.9919 (0.0782)**	0.0224 (0.0279)
EVT-based MLC	0.9859 (0.0317)	0.0074 (0.0041)	0.9827 (0.0695)	0.0972 (0.0516)
Extended DOC	0.9979 (0.0298)	0.0002 (0.0198)	0.9801 (0.1186)	0.0272 (0.0453)
Proposed method (background)	**0.9994 (0.0124)**	**0.0001 (0.0013)**	0.9912 (0.1056)	**0.0107 (0.0278)**

[정답 데이터와 오답 데이터에 대한 평균 시그모이드 출력 확률값]

결론

- 공정의 치명적인 결함을 야기하는 **중요 알람을 조기에 분류**하는 것은 중요
- 기존의 분류기의 경우 학습 당시 수집된 데이터에 존재하는 클래스에 대해서만 분류할 수 있도록 학습
- 학습하지 않은 알람 클래스에 대해 유연하게 대처하기 위해서는 **학습한 클래스로 오분류하지 않는 모델**이 필요함
  - ✓ 배경데이터를 사용한 **다중 레이블 오픈셋 분류 방법론**을 제안
  - ✓ 최신 오픈셋 분류 방법론과 비교하여 **제안 방법론의 우수한 성능 확인**
- 시시각각 변화하는 **실제 공정의 상황에서 유연하게 대처**할 수 있다는 점에서 공정 관리 분야에 큰 기여를 할 것

감사합니다