

24.02.15

## 다변량 스팀 사용 이상 감지 및 영향 변수의 원인 분석

서울과학기술대학교 데이터사이언스학과

이성호 [sean0310@seoultech.ac.kr](mailto:sean0310@seoultech.ac.kr)

배소희 [shbae2819@g.seoultech.ac.kr](mailto:shbae2819@g.seoultech.ac.kr)

심재웅 [jaewoong@seoultech.ac.kr](mailto:jaewoong@seoultech.ac.kr)

# 데이터

## 이용 데이터

### 제품 2종에 대한 6달간 센서 데이터

df\_ext(2023-03,04)(5123,0385)\_2023-11-16 seoultech

df\_ext(2023-05,06)(5123,0385)\_2023-11-16 seoultech

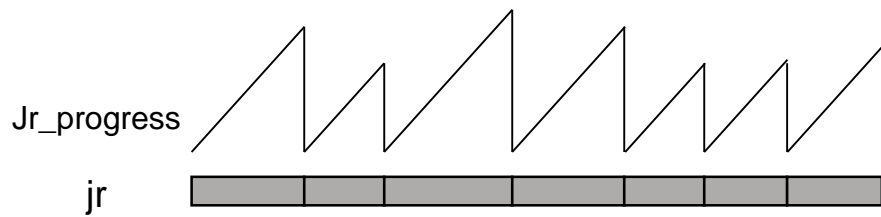
df\_ext(2023-07,08)(5123,0385)\_2023-11-16 seoultech

기간 : 2023-03-02 08:00:00 ~ 2023-08-27 03:00:00 (분)

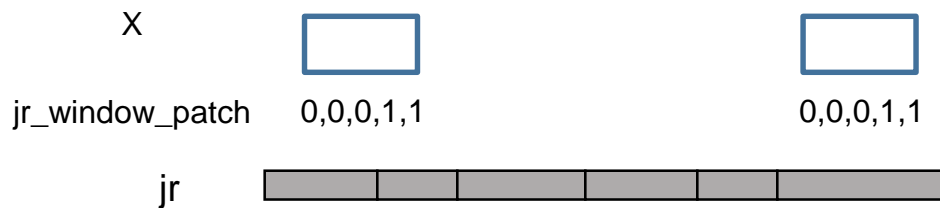
date	날짜
tg	Sensor (38개)
stop	공정 분석값 0 : 가동 1 : 중지 이벤트 발생 2 : 중지 복구
jr	단위 공정값 / 제품 생산 주기 (생산품 번호)

### 이용 데이터 전처리

- jr\_progress : jr을 기준으로 시간에 따른 정수를 새로운 변수로 추가



- jr\_window\_patch : window 사이즈 만큼의 데이터 패치 후, 예측하고자 하는 시점의 jr과 다른지 같은지 표기하는 새로운 변수 추가



# 데이터

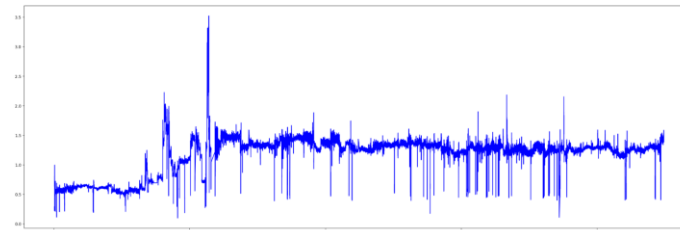
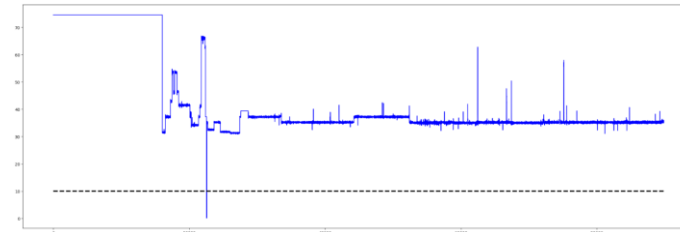
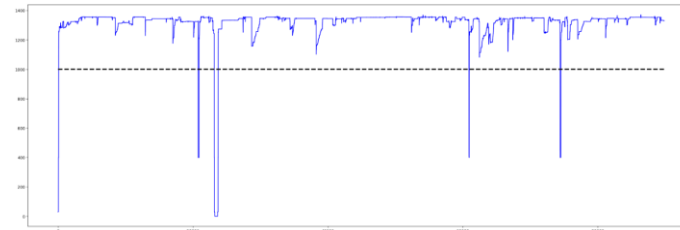
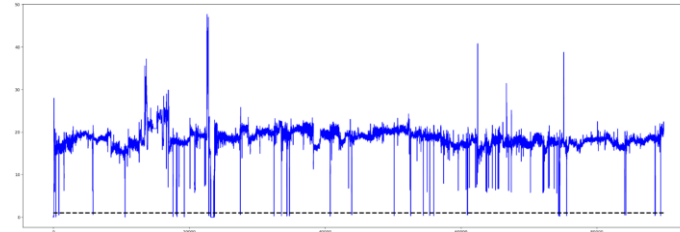
- 이용 데이터
    - 이용 데이터 정의
      - **input(X) : tg (38개 sensor 데이터), 공정진행도(jr\_progress) , 공정변화도(jr\_window\_patch)**
        - Window size : 30
      - **output(y) : 5분 후의 **ei계산값**  $tg04/(tg02*tg03*0.0003)$**
    - Data split
      - Train/Test = 80:20
- Train : 2023-03-02 08:00:00 ~ 2023-07-16 12:47:00 (70440)
- Test : ~ 2023-08-27 03:00:00 (17598)

# 데이터 이해 및 전처리

## ● 이용 데이터 처리

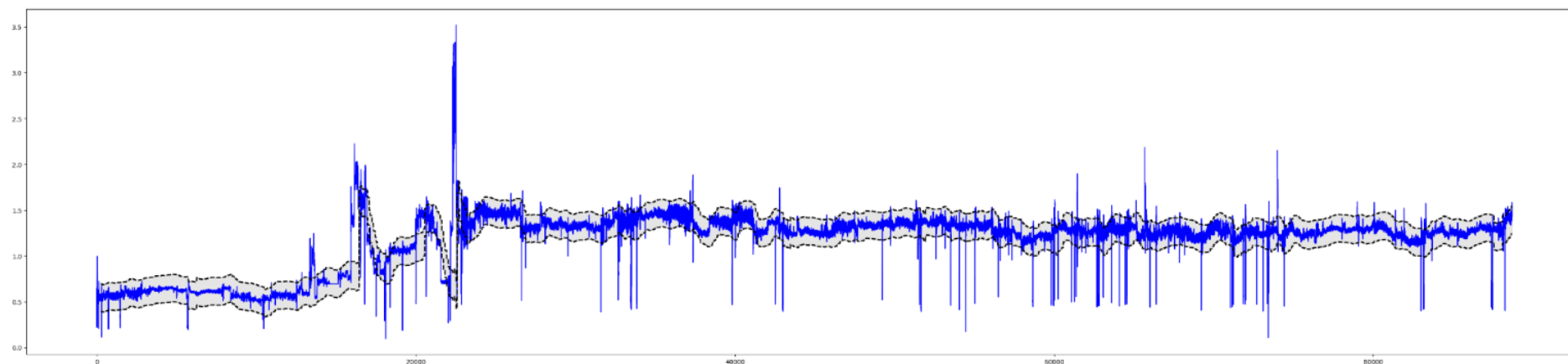
- tg04  
tg02 값 중 1 이하의 값 제외(735개의 index)
- tg03  
tg03 값 중 1000 이하의 값 제외(959개의 index)
- tg02  
tg02 값 중 10 이하의 값 제외(54개의 index)
- 결과  
총 1,131개의 row가 삭제됨

```
count    88709.000000  
mean      1.174082  
std       0.305062  
min       0.096630  
25%      1.158493  
50%      1.274723  
75%      1.343347  
max       3.523615  
Name: ei, dtype: float64
```



# UCL & LCL 설정(이상치 정의)

- 이전 5개의 jr에 대한 ei값의 누적 평균  $\pm 0.15$
- 전체 시계열
  - 관리 한계선(UCL,LCL) 내부 : class 0 (82668)
  - LCL 미만 : class 1 (2794)
  - UCL 초과 : class 2 (3247)



# 1D CNN

## ● Experiment setting

- Epoch : 100
- optimizer : Adam(lr=1e-4)
- 학습 데이터셋에서 각 클래스가 차지하는 비율의 역수를 사용하여 loss weight 설정
- Layer : 1D Conv layer(16->32->64->128) / kernel=3),  
Linear layer (256 -> 32 -> 3)

Confusion Matrix\_1D CNN\_classification

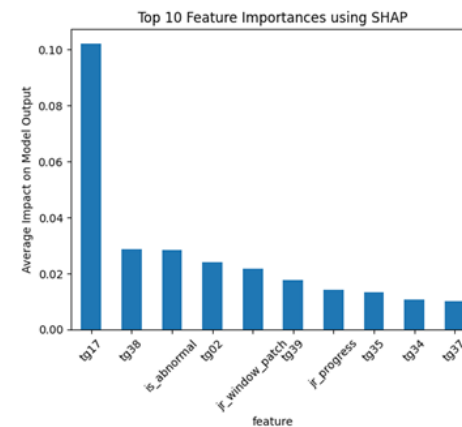
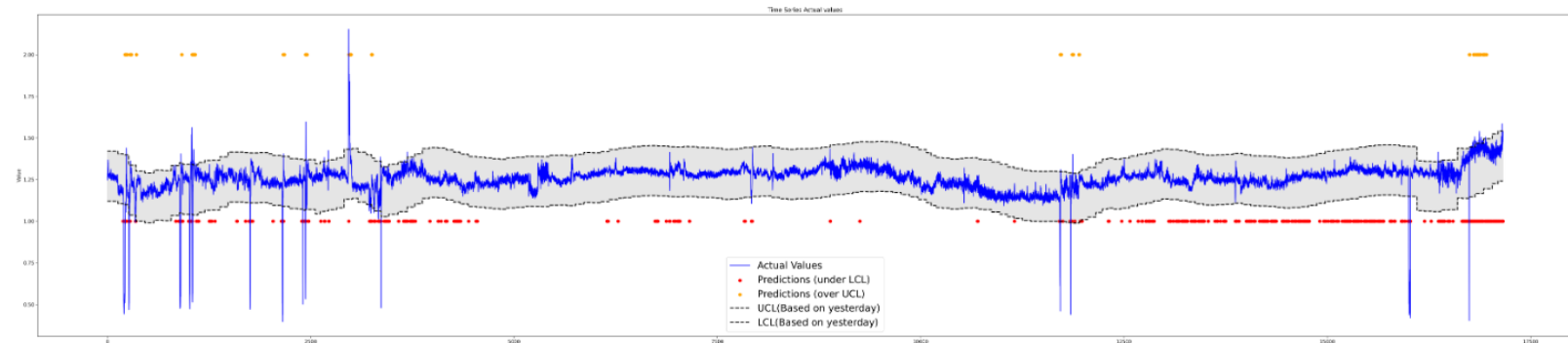
Predicted label \ True label	ok	Under LCL	Over UCL
ok(n=16907)	15308	1482	117
Under LCL(n=138)	16	113	9
Over UCL(n=115)	42	33	40

## ● Result

- Acc : 0.9010
- F1 : 0.4538

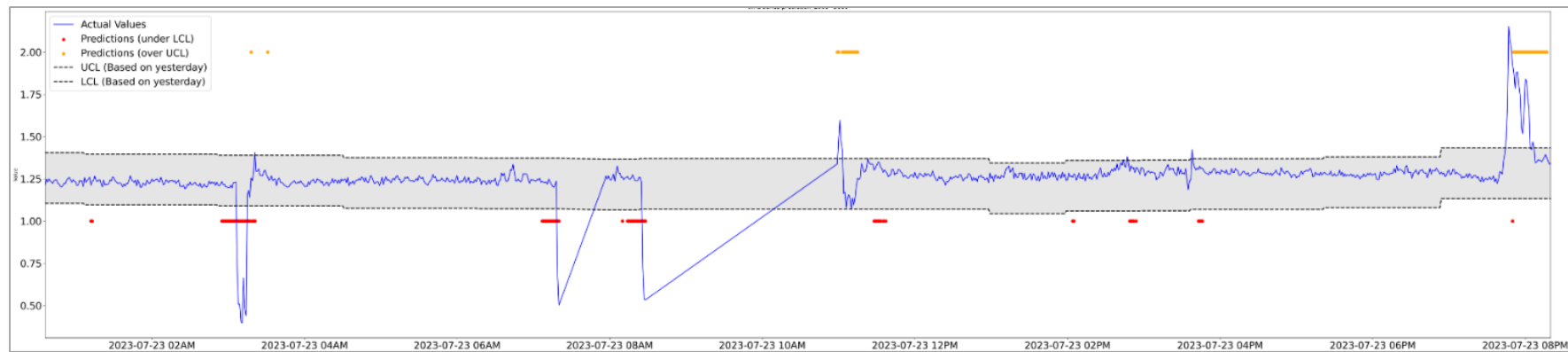
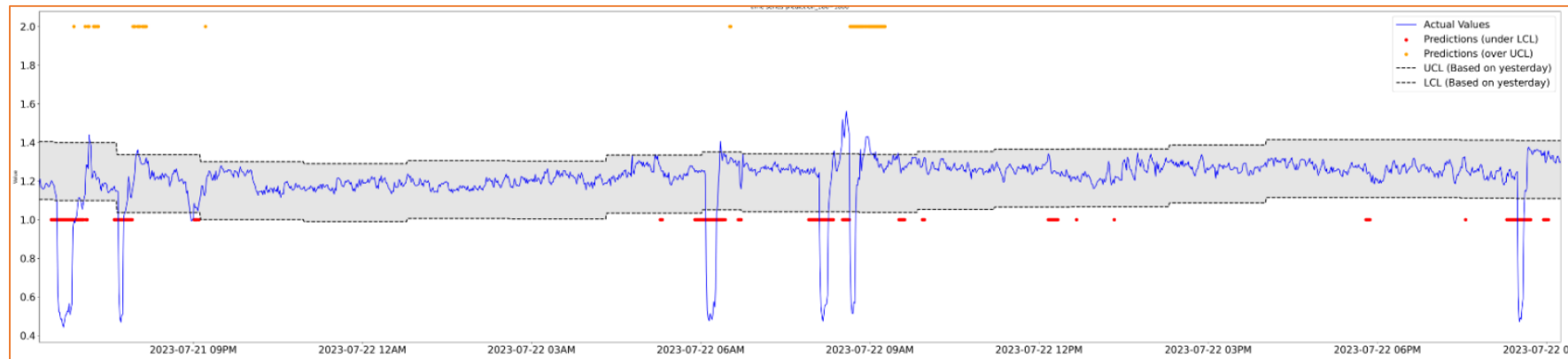
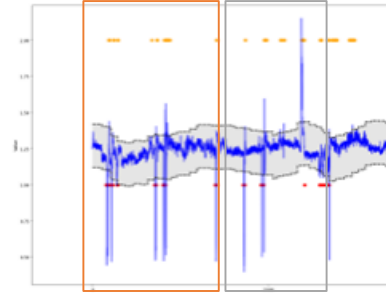
## ● 변수 별 중요도 해석

- tg17(스팀 누적값)
- tg38(설비 AE 속도)
- is\_abnormal(ei label값)



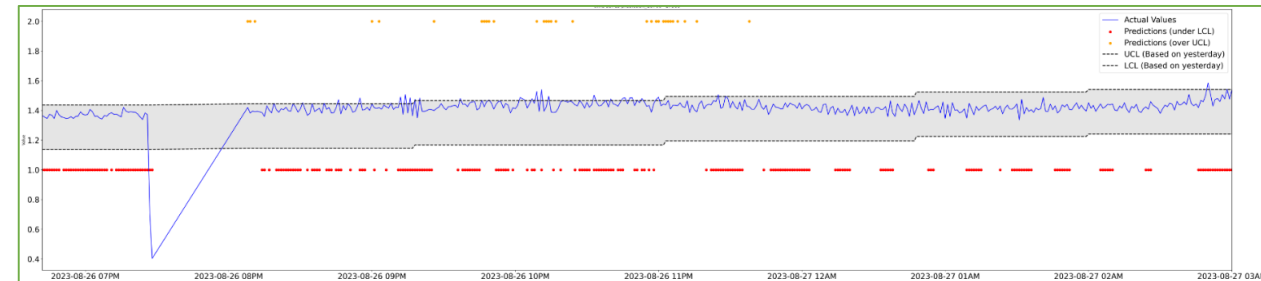
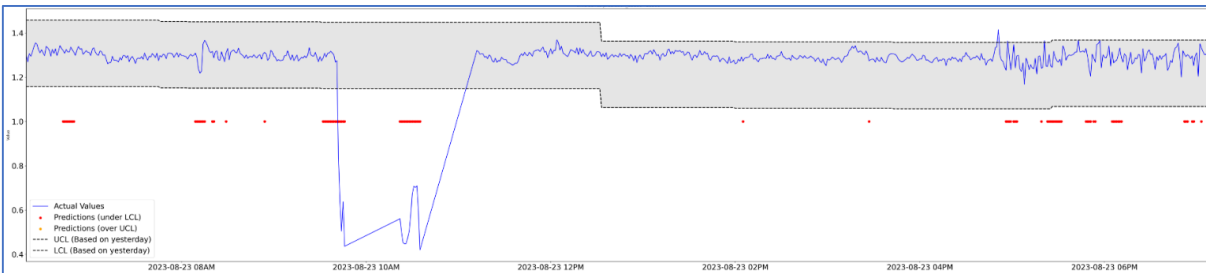
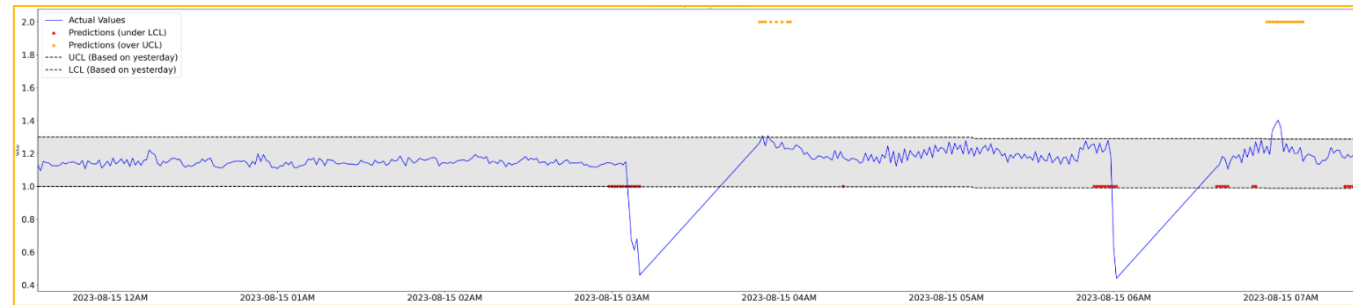
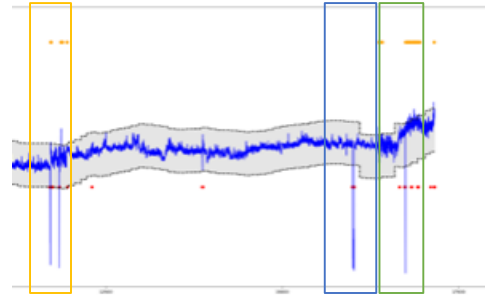
# 1D CNN

- 모델이 abnormal이라고 분류하는 index



# 1D CNN

- 모델이 abnormal이라고 분류하는 index





# LSTM

## ● Experiment setting

- Epoch : 100
- optimizer : Adam(lr=1e-4)
- 학습 데이터셋에서 각 클래스가 차지하는 비율의 역수를 사용하여 loss weight 설정
- Layer : lstm layer(hidden=256, layer=6) + attention layer

Confusion Matrix\_LSTM

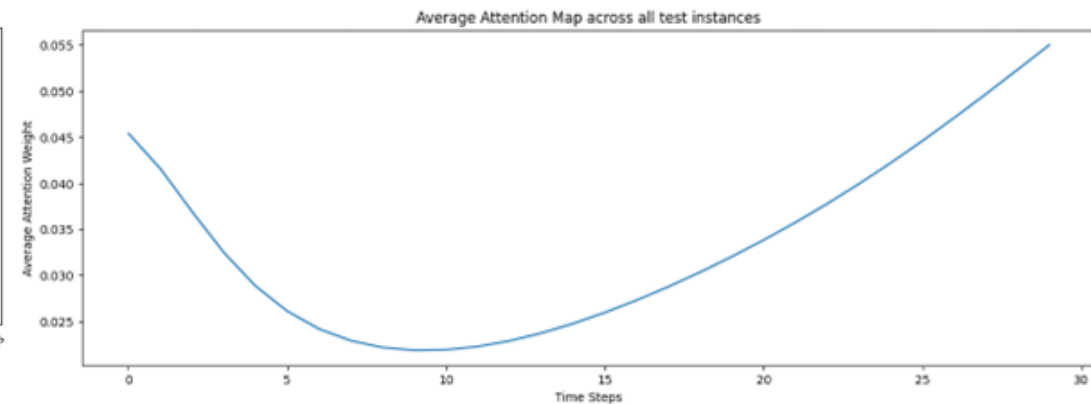
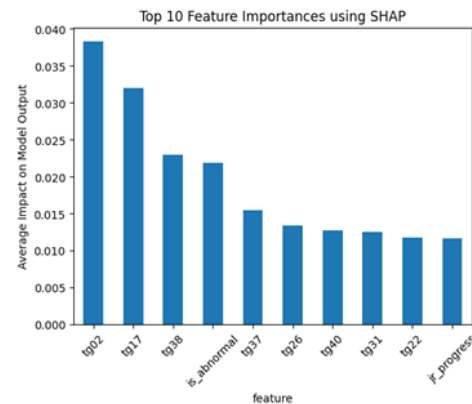
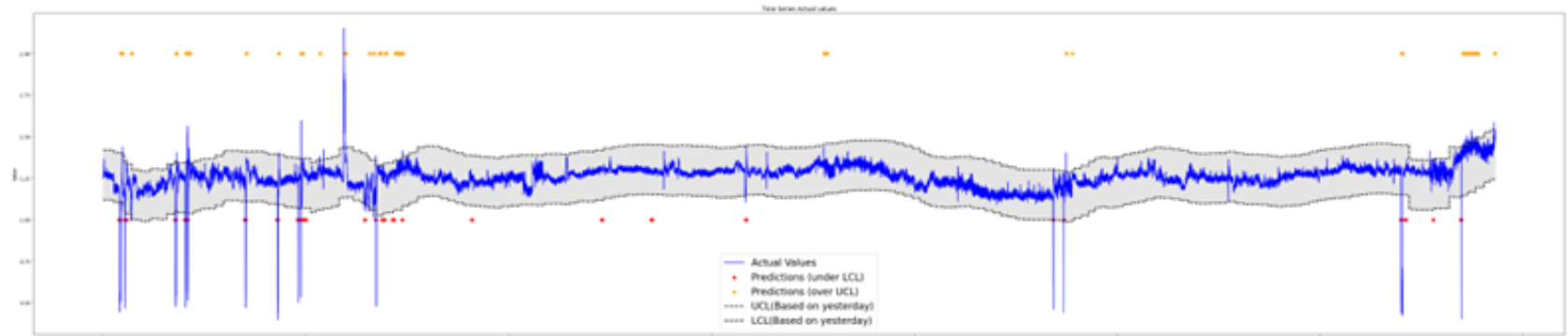
Predicted label \ True label	ok	Under LCL	Over UCL
ok(n=16477)	16390	42	45
Under LCL(n=287)	213	60	14
Over UCL(n=396)	304	36	56

## ● Result

- Acc : 0.9619
- F1 : 0.4945

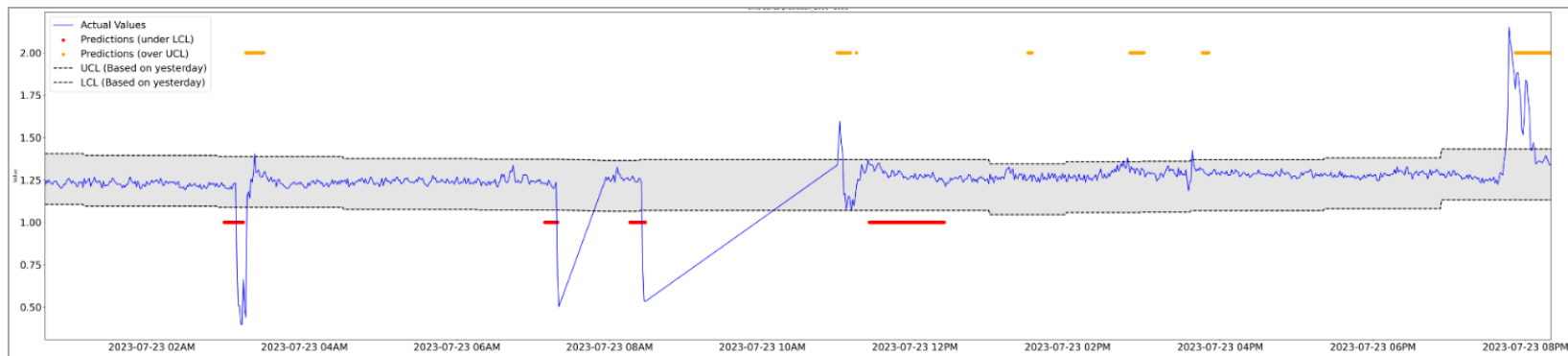
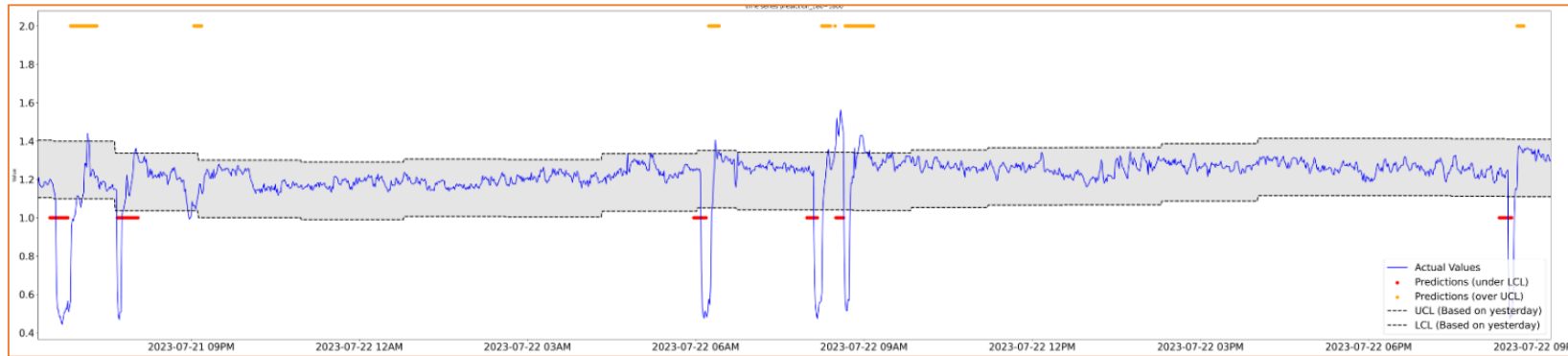
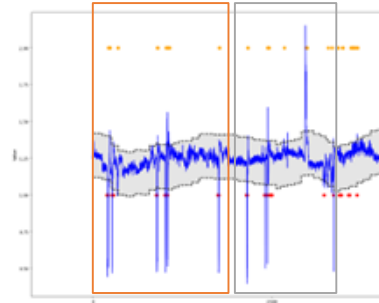
## ● 변수 별 중요도 해석

- tg02(종이별 측정 무게)
- tg17(스팀 누적값)
- tg38(설비 AE 속도)



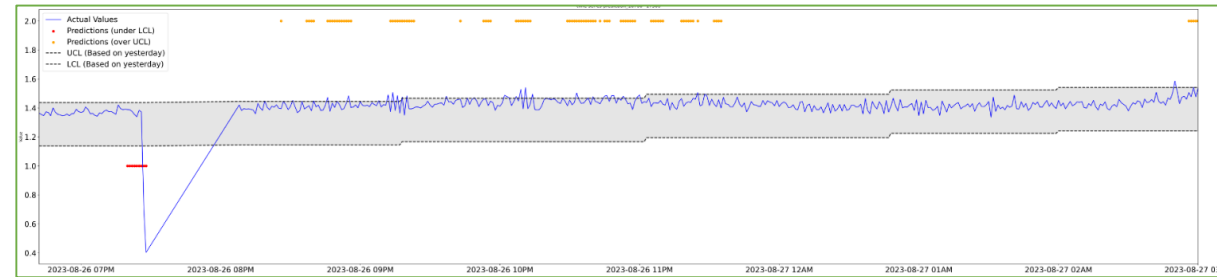
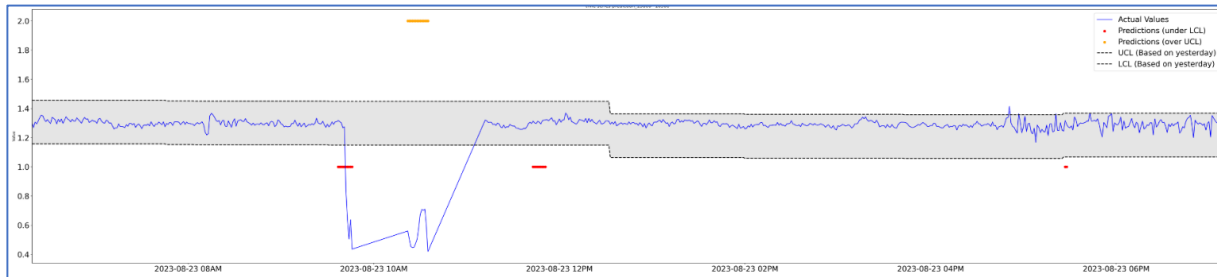
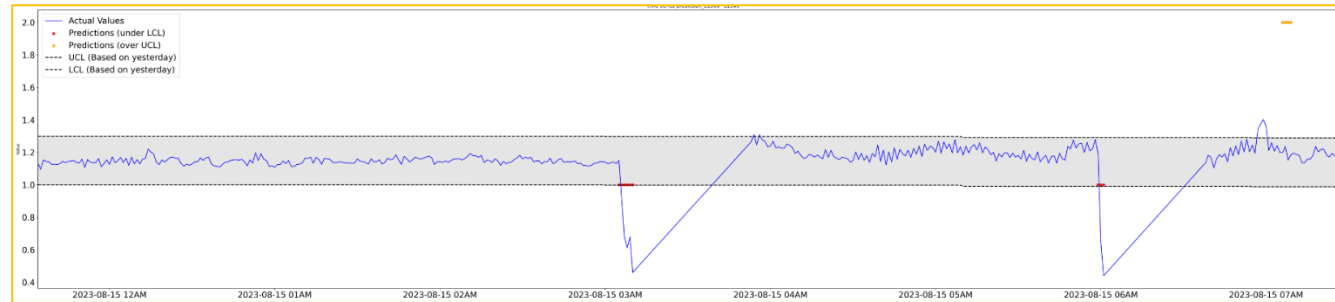
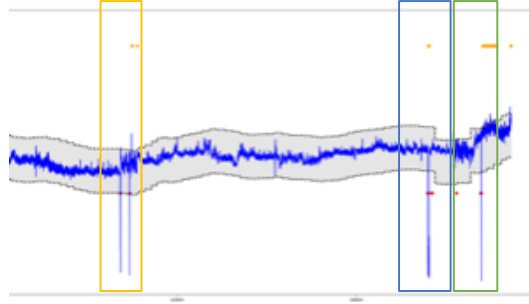
# LSTM

- 모델이 abnormal이라고 분류하는 index



# LSTM

- 모델이 abnormal이라고 분류하는 index



# IMV-LSTM

## ● Experiment setting

- Epoch : 100
- optimizer : Adam(lr=1e-3)
- Lstm layer 노드 : 32
- 학습 데이터셋에서 각 클래스가 차지하는 비율의 역수를 사용하여 loss weight 설정

## ● Result

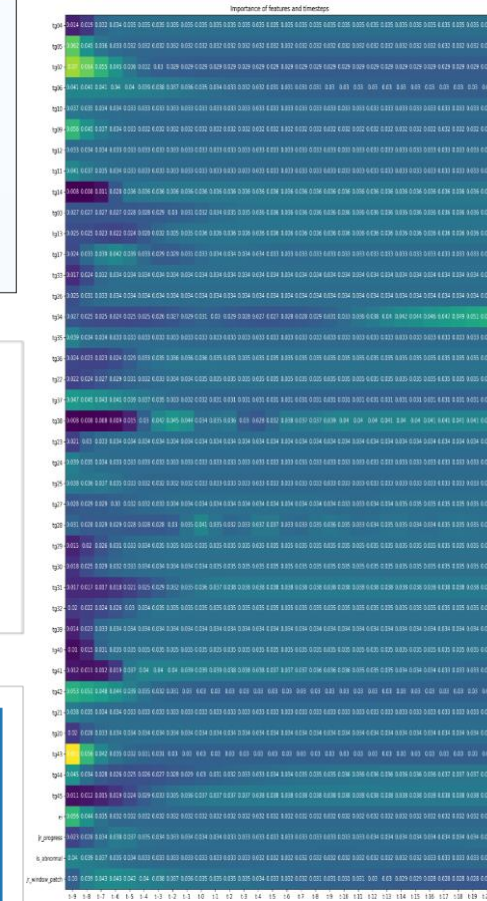
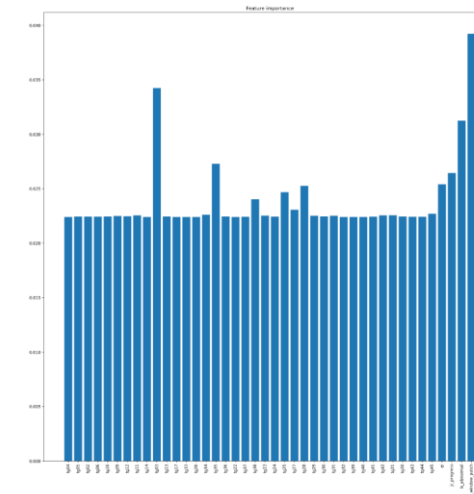
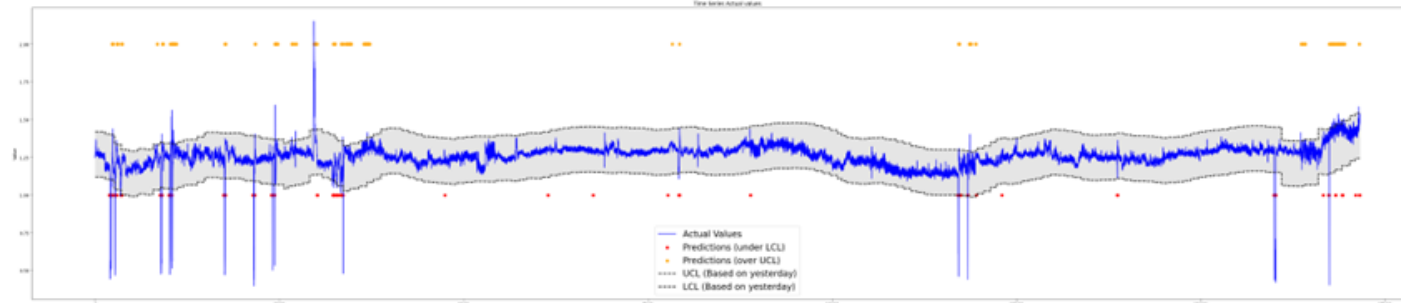
- Acc : 0.9501
- F1 : 0.5368

## ● 모델 해석(Attention Map)

- jr\_window\_patch(공정변화도)
- tg03(끝단 설비 속도)
- is\_abnormal(ei label값)
- tg 35(설비S1 용액 높이)
- jr\_progress(공정진행도)

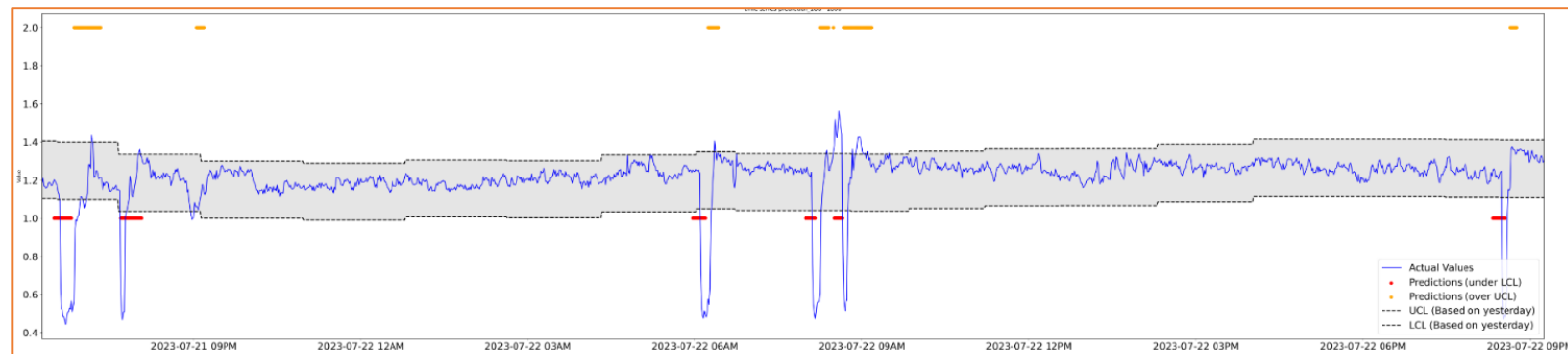
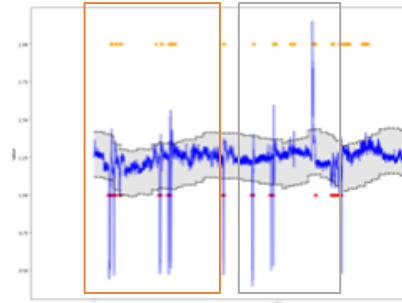
Confusion Matrix\_IMV-LSTM

	ok(n=16316)	Under LCL(n=348)	Over UCL(n=496)
Predicted label	16232	47	37
ok	16232	47	37
Under LCL	263	78	7
Over UCL	412	13	71



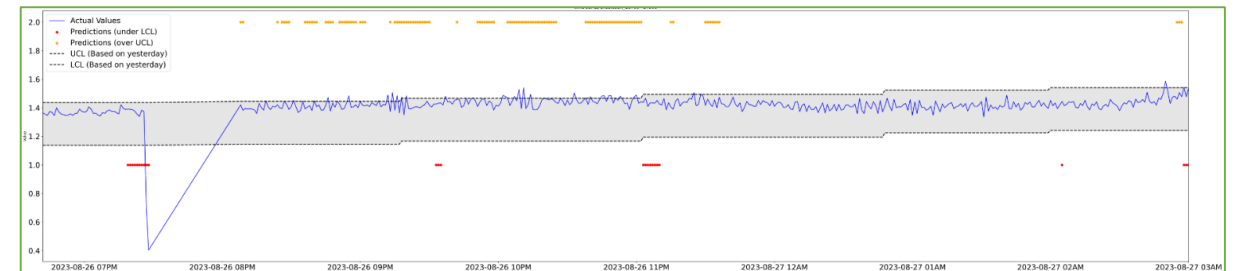
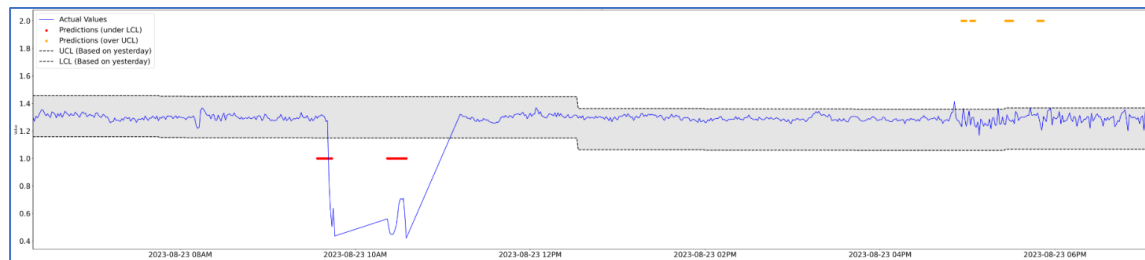
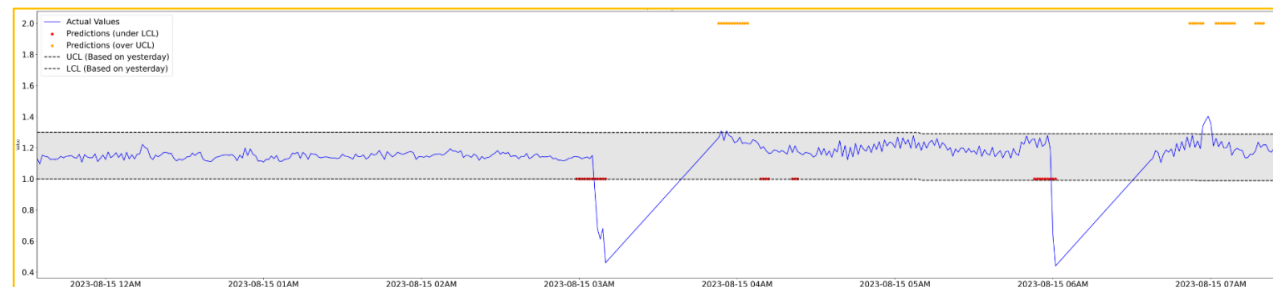
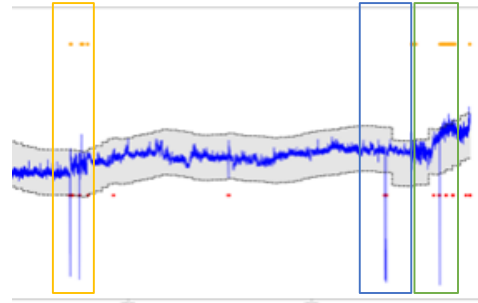
# IMV-LSTM

- 모델이 abnormal이라고 분류하는 index



# IMV-LSTM

- 모델이 abnormal이라고 분류하는 index



# 전체 모델 결과표

## ● 1D CNN

Regression			
	precision	recall	F1 score
Under LCL	0.3986	0.4365	0.4167
Over UCL	0.5391	0.2081	0.3002
Accuracy			0.9748

Classification			
	precision	recall	F1 score
Under LCL	0.8188	0.0694	0.1280
Over UCL	0.3478	0.2847	0.2847
Accuracy			0.9010

- 회귀에서 높은  
정확도이나 분류의  
정확도는 낮음  
- 분류의 F1 score 매우  
낮음

## ● LSTM

Regression			
	precision	recall	F1 score
Under LCL	0.6087	0.4200	0.4970
Over UCL	0.2087	0.4000	0.2743
Accuracy			0.9832

Classification			
	precision	recall	F1 score
Under LCL	0.4348	0.2091	0.2824
Over UCL	0.487	0.1414	0.2192
Accuracy			0.9619

- 회귀의 Under UCL의  
경우 높은 정밀도  
- 전반적으로 낮은 F1  
score

## ● IMV LSTM

Regression			
	precision	recall	F1 score
Under LCL	0.5072	0.4930	0.5000
Over UCL	0.2870	0.3667	0.3220
Accuracy			0.9843

Classification			
	precision	recall	F1 score
Under LCL	0.5652	0.2241	0.3210
Over UCL	0.6174	0.1431	0.2324
Accuracy			0.9546

- 회귀에서의 높은  
정확도와 Under UCL의  
가장 높은 재현율  
- 분류의 F1 score 가장  
높음

→ IMV LSTM이 회귀와 분류 작업 모두에서 가장 균형 있는 성능

감사합니다