

시계열 공정 데이터를 활용한 장비 이상 조기 탐지 자동화 시스템 구현

Manufacturing Process Data Anomaly Detection

김재욱 | 김재운 | 김민석 | 하연진

index

01 프로젝트 개요

- 01 배경 및 필요성
- 02 프로젝트 목적

02 데이터 전처리

- 01 데이터 소개 및 병합
- 02 데이터 Labelling
- 03 학습 · 평가 데이터 준비

03 EDA

- 01 변수 분포
- 02 상관관계

04 모델링

- 01 LSTM-AE
- 02 Isolation Forest

05 효과 및 의의

- 01 기대효과
- 02 개선점 및 의의

01

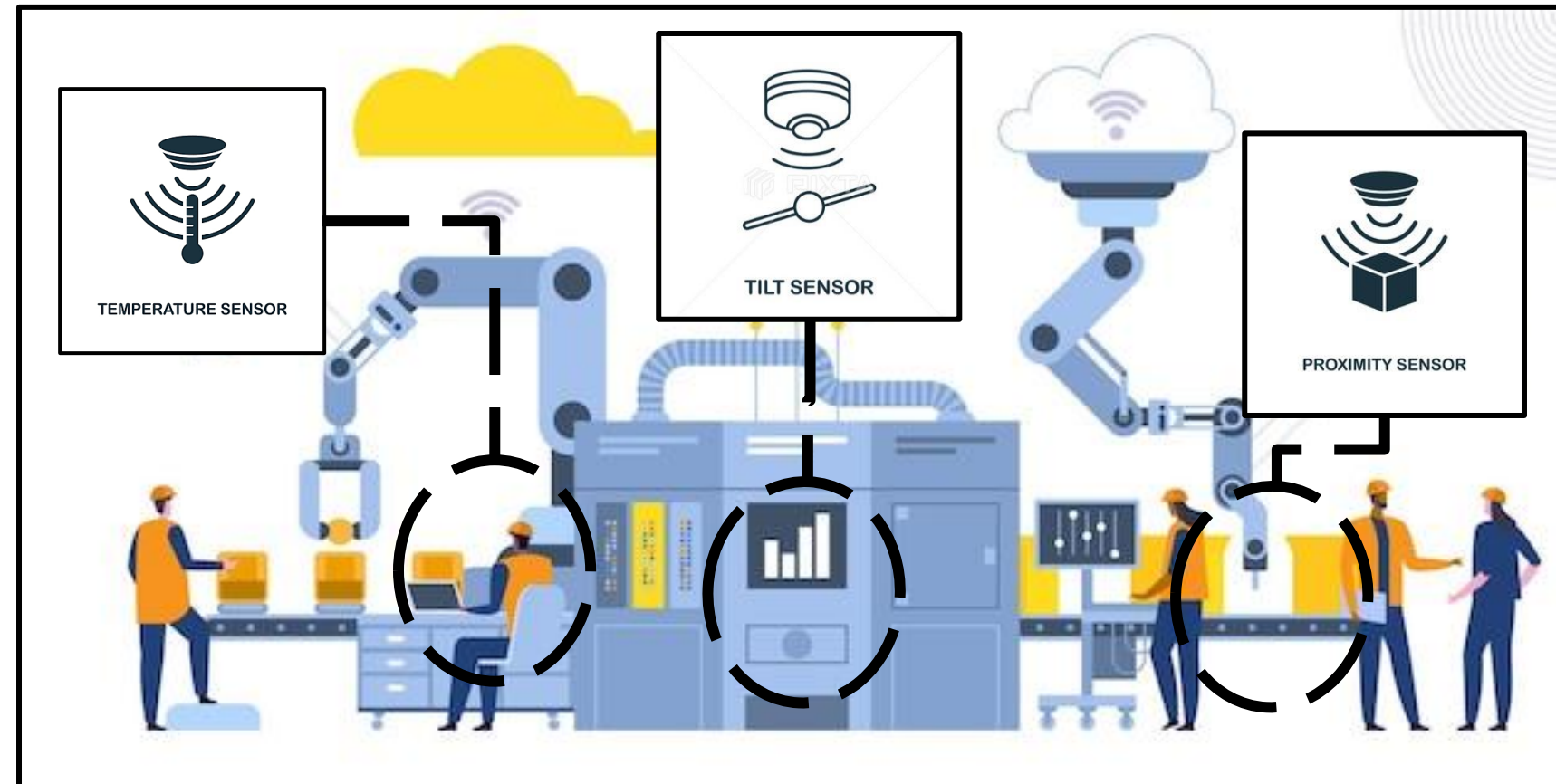
프로젝트 개요

01 배경 및 필요성

02 프로젝트 목적

배경 및 필요성

프로젝트 개요



공정 데이터 분석 (PDA: Process Data Analytics)

설비 상태 정보, 작동 기록 및 기타 데이터를 분석하여 제조 초기 단계에서 품질 또는 생산성 저하를 감지 할 수 있는 프로그램

배경 및 필요성

프로젝트 개요

SK하이닉스, 인공지능(AI) 솔루션 도입...반도체 공정 효율성 개선

신미정 | 입력 2023.01.10 18:25 | 수정 2023.01.10 18:26

댓글 0

가

SK하이닉스가 반도체 공정에 인공지능(AI) 솔루션을 도입해 생산 운영 효율 및 수율 개선에 나섰다.

SK하이닉스는 지난달부터 제조 공정 결과를 센서 데이터를 활용해 예측하는 가상 계측 AI 솔루션 '파놉테스 VM(Panoptes Virtual Metrology)'을 지난달부터 양산 팹에 도입해 사용하고 있다고 10일 밝혔다.

금속 산업의 디지털화, '선택'이 아닌 '필수'의 시대

게르트 크라우제 | 등록 2023.01.29 11:13:08

URL복사

최근 전 세계적으로 화두가 되고 있는 '디지털화'의 물결은 금속 산업에서도 요동치고 있다. 금속 산업의 디지털화의 노력은 '수익 극대화'와 '탈탄소화 이룩'이라는 두 마리 토끼를 잡기 위한 과정이다.

디지털화의 핵심은 제품과 기계 데이터를 다양한 공정 단계에 거쳐 분석하는 것이다. 정확하게 예측하고, 오차 허용치를 철저히 준수하는 것. 그것이 디지털화의 핵심이다. 금속 산업은 디지털화를 통해 확고한 산업 경쟁력을 제고할 수 있다.

금속 산업의 새로운 어젠다로 떠오르고 있는 디지털화는 금속 업계의 새로운 먹거리다.

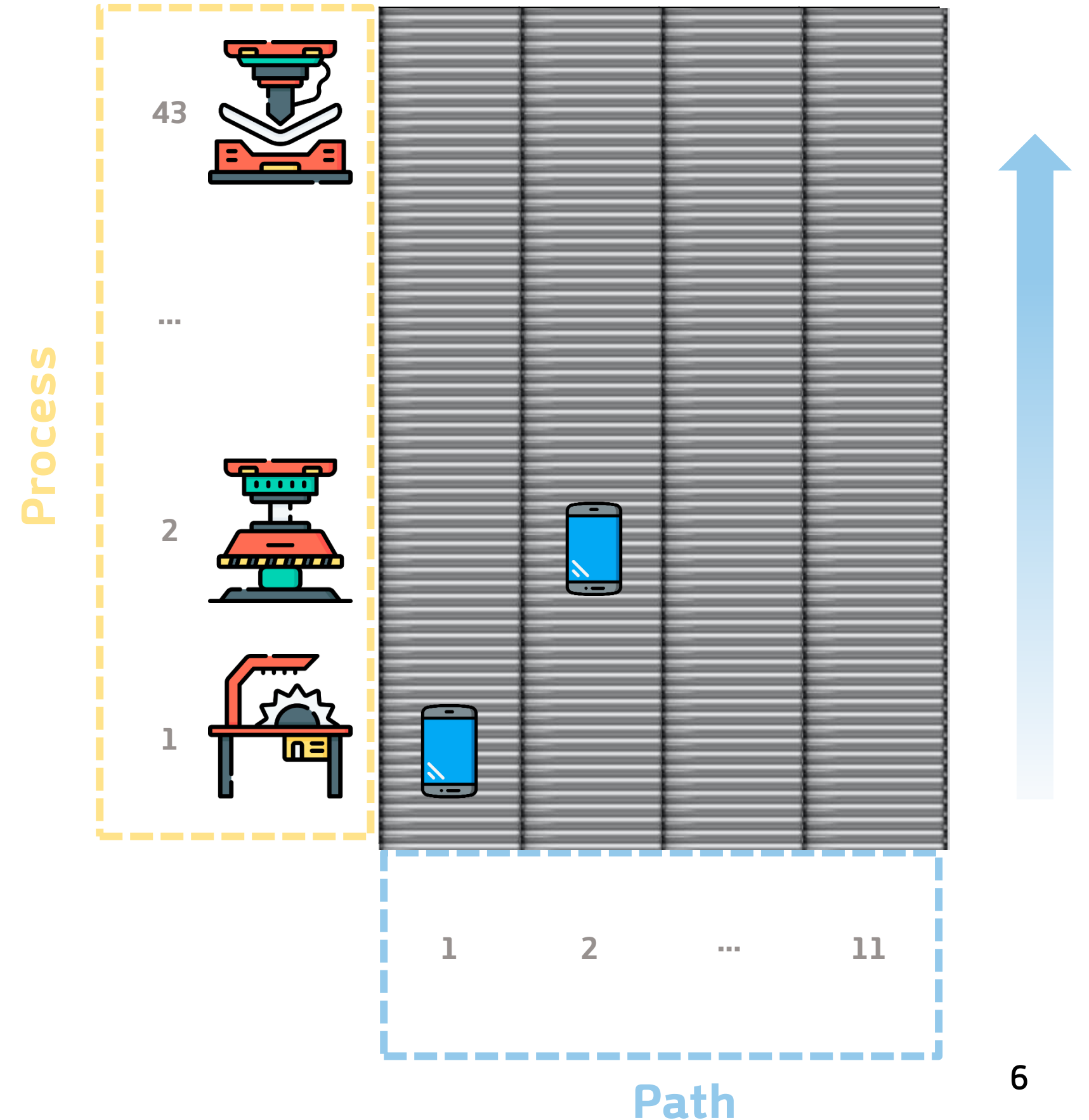
공정 데이터의 분석, 하나의 **트렌드**로 자리잡다

열풍 건조 공정 과정

- 열풍 건조 : 도금처리 이후 피막과 안정제가 금속 표면에 안정적으로 달라붙을 수 있게 하는 공정
- 제품은 11개의 Path 각각에서 43개의 Process를 거치며 1개의 Process는 3분이 소요된다.

공정상의 이슈 사항 (Pain Point)

- ★ 육안을 통한 양품선별로 품질의 균일성을 확보 어려움
- ★ 열풍 건조 공정의 높은 전력 소모 구조로 인해 설비의 고장 발생
- ★ 공정 도중에 장비의 이상 여부 확인 불가
- ★ 장비 고장 -> 내부 온도 상승 -> 사고 유발 및 수율 감소



프로젝트 목적

프로젝트 개요

“

장비 이상 조기 탐지 및 공정 제어 자동화 시스템

열풍건조 설비의 내부 온도, 코일의 전류, 모터의 진동 패턴 등을
파악하여 기존의 정상 상태와의 비교 분석 및 향후 설비 상태에 대한
조기 감지로 공정 제어(지속·종료) **자동화 시스템**을 구축한다.

02

데이터 전처리

01 데이터 소개 및 병합

02 데이터 Labeling

03 학습 · 평가 데이터 준비

데이터 소개 및 병합

데이터 전처리

날짜별 공정 데이터

9월 6일, 33번 Process 모두 Error!

Index	Process	Time	Temp	Current	Date
1	1	오후 4:24:03.0	75.139142	1.61	2021-09-06
2	1	오후 4:24:08.0	76.660421	1.53	2021-09-06
3	1	오후 4:24:13.0	77.177660	1.701	2021-09-06
4	1	오후 4:24:18.0	76.586434	1.736	2021-09-06
5	1	오후 4:24:23.0	77.877103	1.748	2021-09-06

Process : process 추적을 위해, 동일 process에 동일 숫자 부여

Temp : 열풍건조 설비 내 공정 온도 측정 값

Current : 열풍건조 설비 내 공정 전압 측정 값

통합 날짜 에러 데이터

9월 6일, 2번 Path, 33번 Process가 Error!

0	1	2	...	10	11
2021-09-06	32	33	...		
2021-09-07	32	33	...		
2021-09-08			...		
2021-09-09	15	16	...		
2021-09-10	32	28	...		

0 : 날짜

1-11 : 에러 발생 로트

Process (1-43)

데이터 Labeling

데이터 전처리

COPkemp-abh-sensor.csv

Index	Process	Time	Temp	Current	Date
1	1	오후 4:24:03.0	75.13914228	1.61	2021-09-06
2	1	오후 4:24:08.0	76.66042142	1.53	2021-09-06
3	1	오후 4:24:13.0	77.17766014	1.701	2021-09-06

Error Lot list.csv

0	1	2	3	4	5
2021-09-06	32	33	20	21	22
2021-09-07	32	33	34		
2021-09-08					



trainset.csv

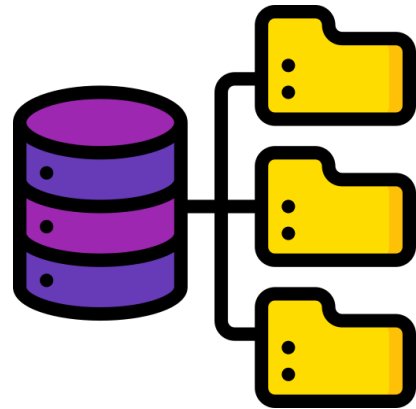
Index	Process	Time	Temp	Current	Date	NG
1	1	오후 4:24:03.0	75.13914228	1.61	2021-09-06	0
2	1	오후 4:24:08.0	76.66042142	1.53	2021-09-06	0
3	1	오후 4:24:13.0	77.17766014	1.701	2021-09-06	0

학습 · 평가 데이터 준비

데이터 전처리

STEP 01

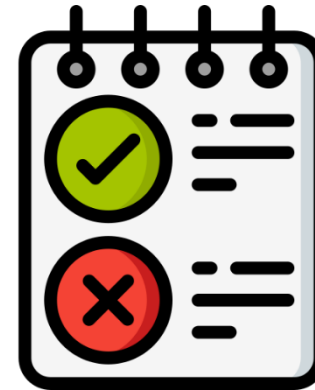
Train_set / Valid_set
정상 데이터



- 정상 데이터는 Train, Valid로 활용
- Train_set : Valid_set = 8 : 2
- Valid_set은 **Threshold** 설정에 활용

STEP 02

Test_set
정상/비정상 데이터



- Error 데이터는 테스트 데이터로 활용
- NG 데이터와 1:1 비율로 정상 데이터도 추출하여 Test_set 구성

STEP 03

Scaling
Process 마다 진행



- Standard Scaler -> 평균0, 표준편차1

03

EDA

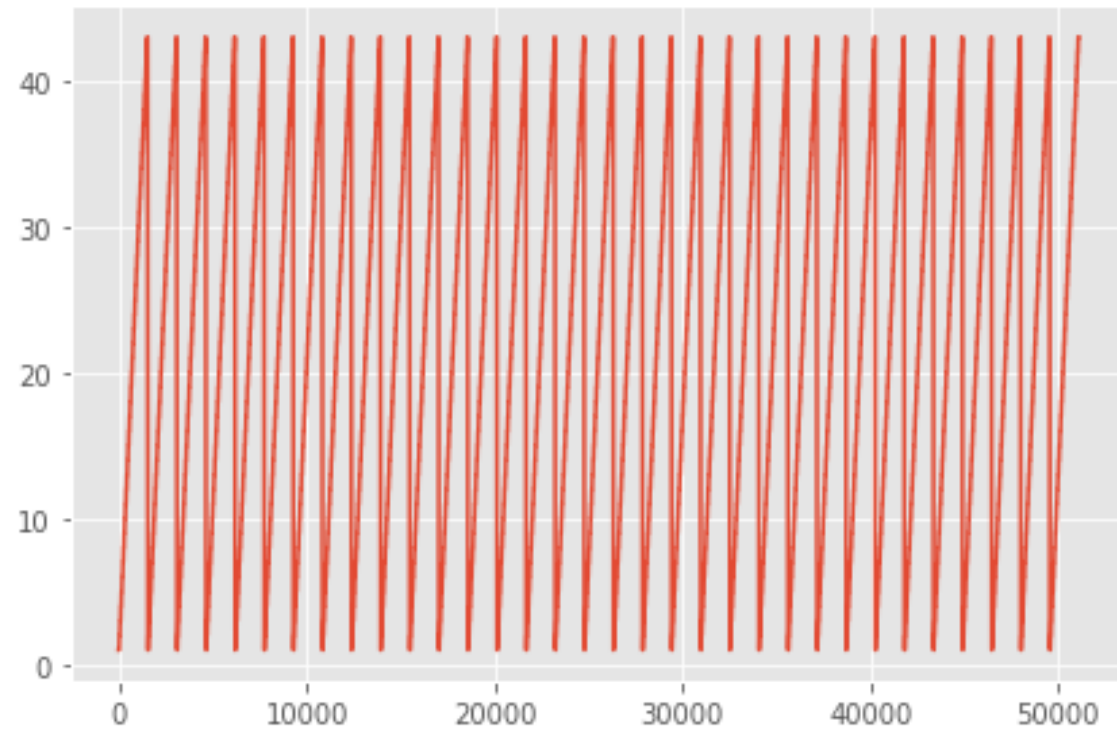
01 변수 분포

02 상관관계

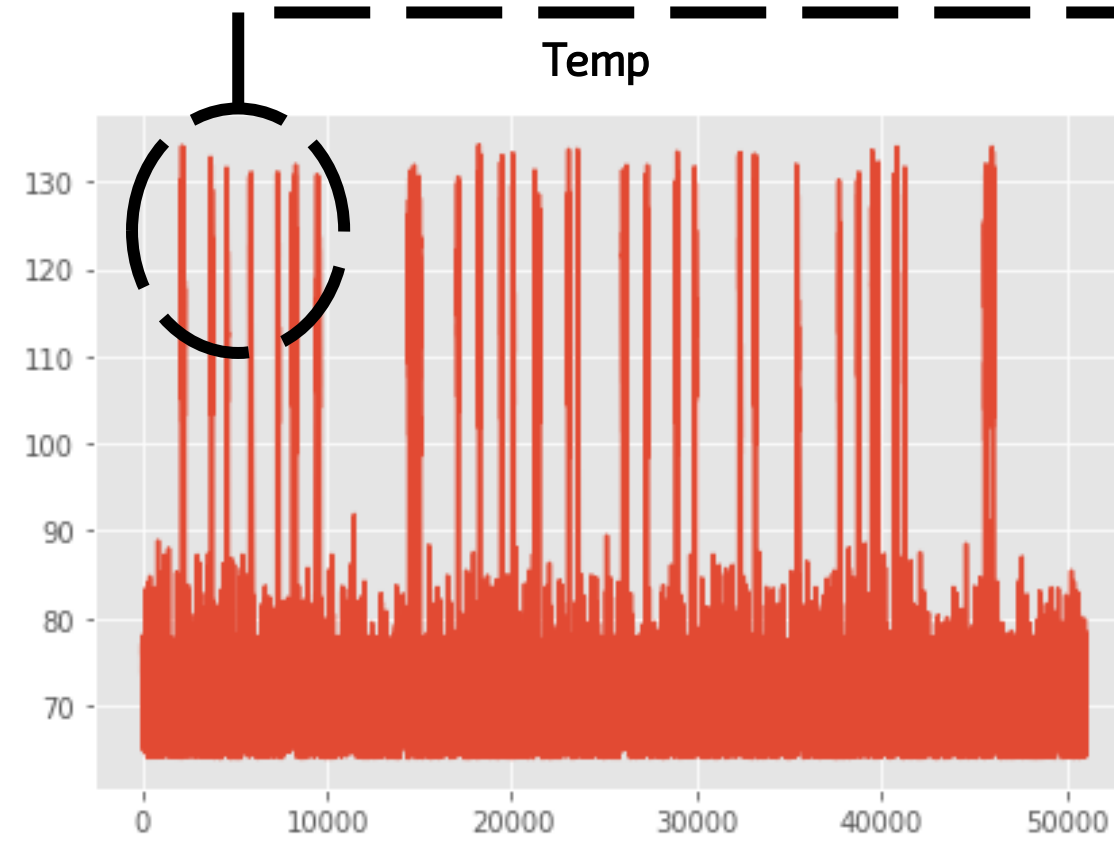
변수 분포

EDA

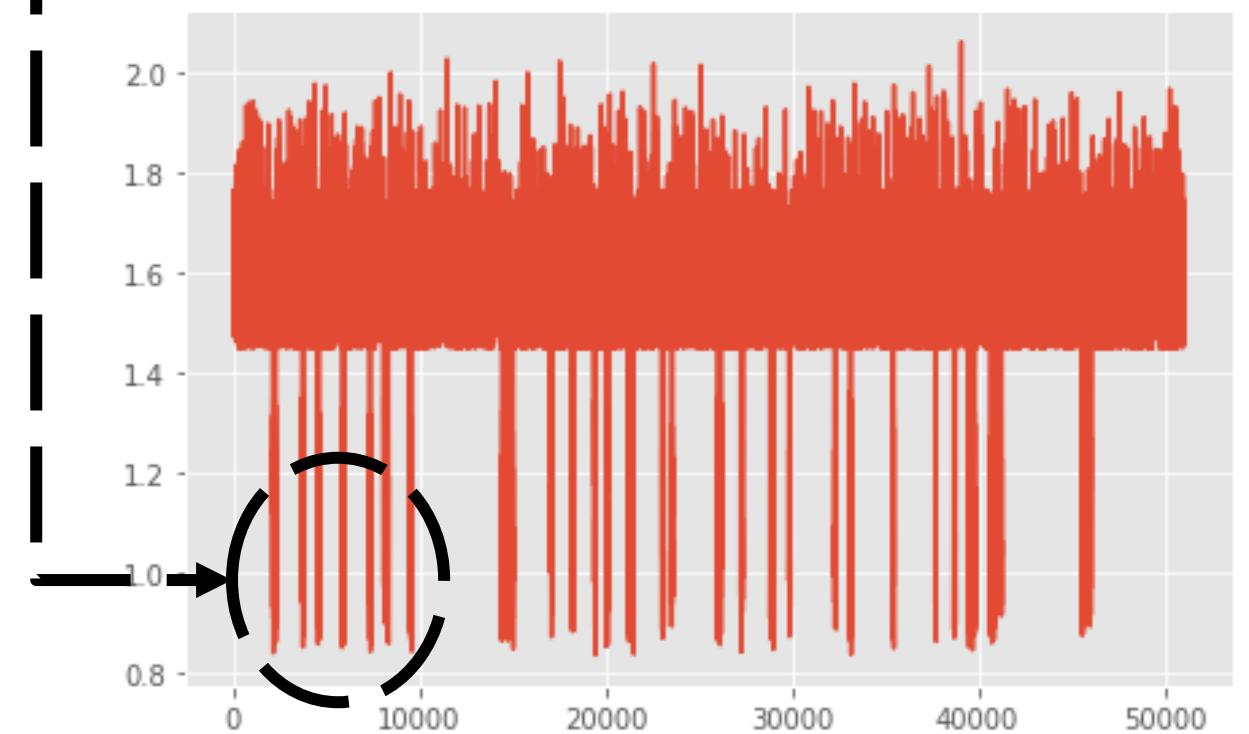
Process



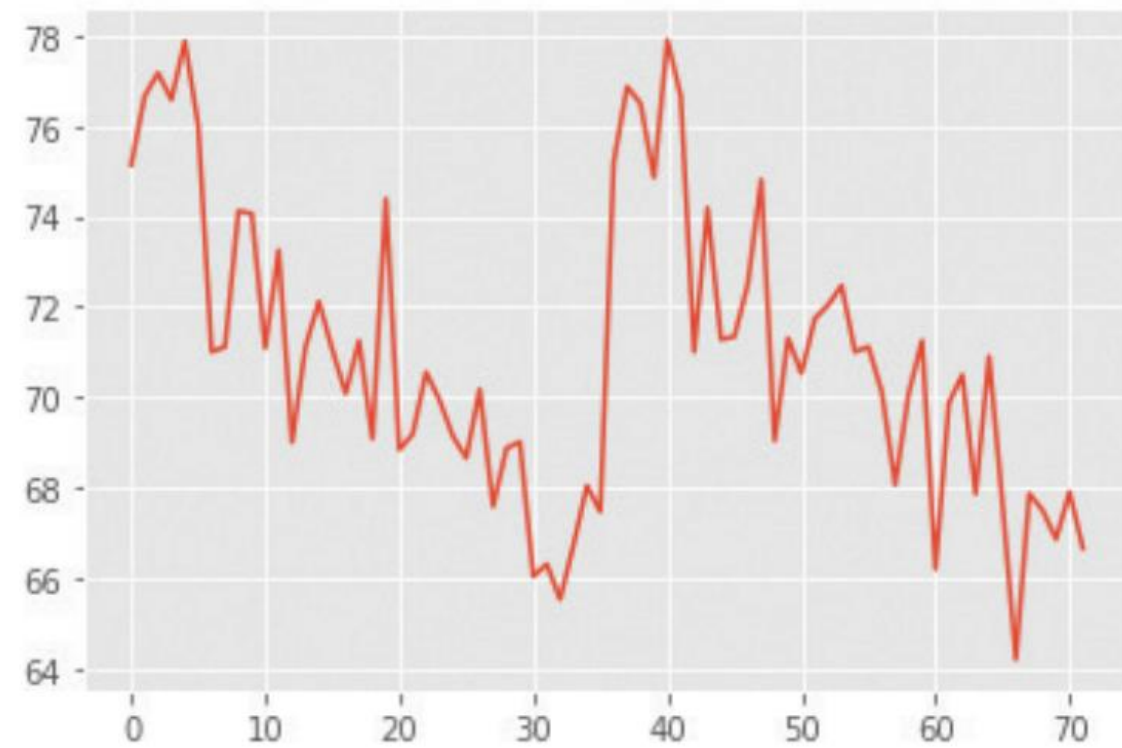
Temp



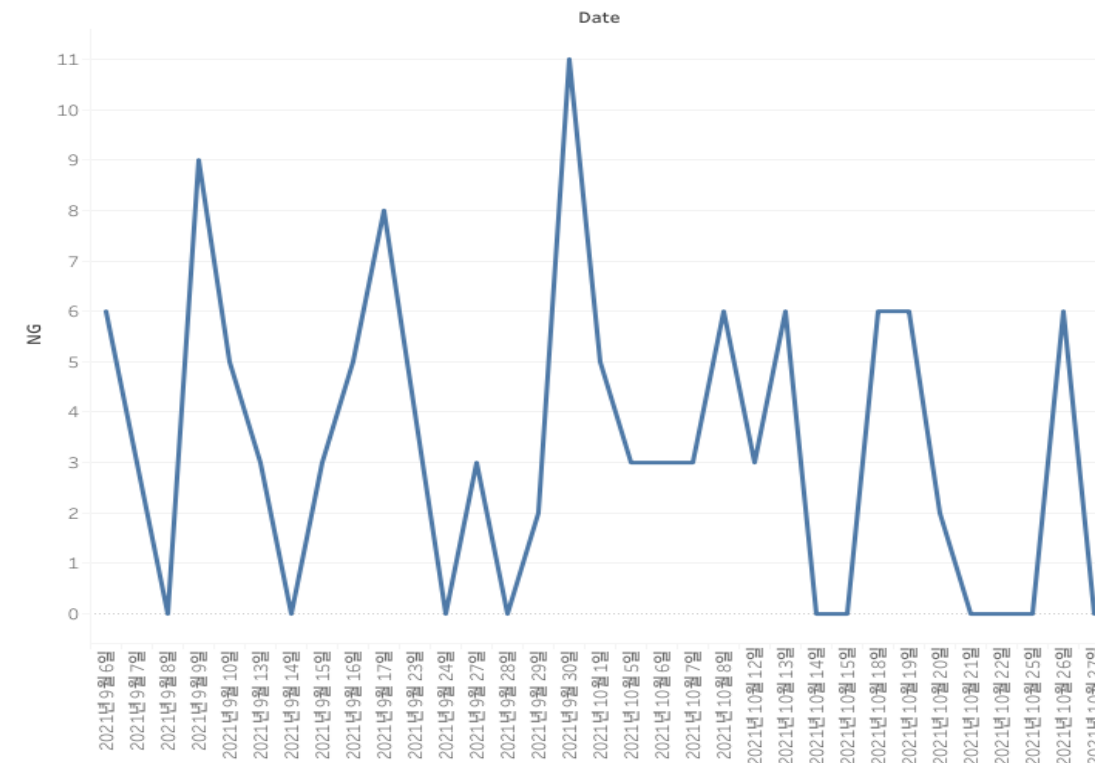
Current



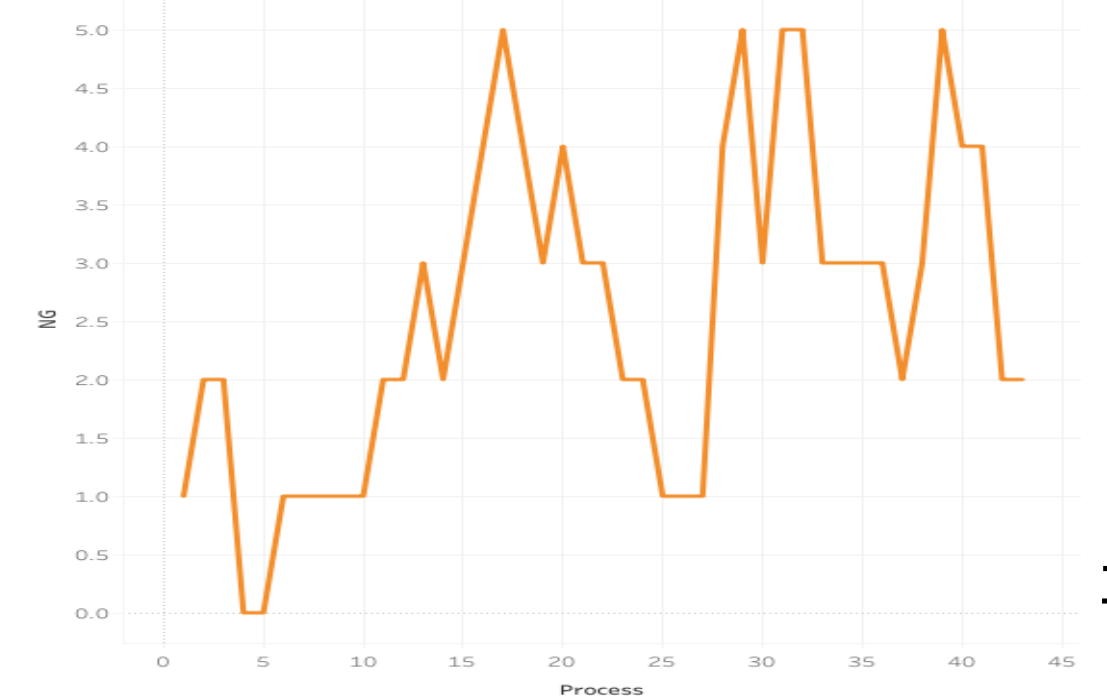
72개의 Temp 데이터를 라인 그래프로



NG 합 by Date



NG by Process



상관관계

EDA

Temp & Current

	Temp	Current
Temp	1.000000	-0.733613
Current	-0.733613	1.000000

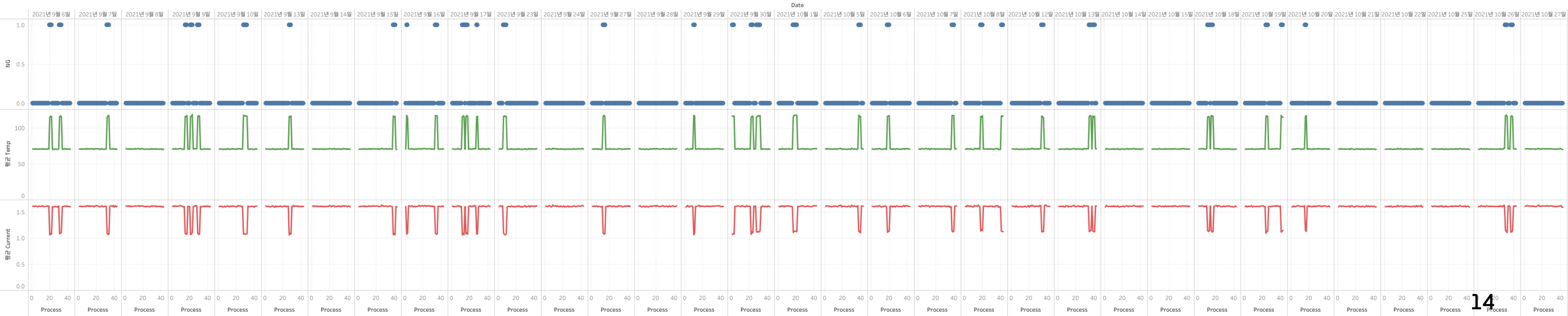


Temp와 Current 데이터는 강한 음의 상관관계를 가진다.

r 값이 ± 0.1 이면, 없다고 할 수 있는 상관관계
r 값이 $\pm 0.1 \sim \pm 0.3$ 이면, 약한 양적 상관관계
r 값이 $\pm 0.3 \sim \pm 0.7$ 이면, 양적 상관관계
r 값이 ± 0.7 이상이면, 강한 양적 상관관계

NG & Temp & Current

NG&Temp&Current



04

모델링

01 LSTM-AE

02 Isolation Forest

LSTM-AE

모델링

Time Series Anomaly Detection

35 papers with code • 0 benchmarks • 2 datasets

This task has no description! [Would you like to contribute one?](#)

[Edit](#)

Benchmarks

These leaderboards are used to track progress in Time Series Anomaly Detection

No evaluation results yet. Help compare methods by [submitting evaluation metrics](#).

[Add a Result](#)

Libraries ①

Use these libraries to find Time Series Anomaly Detection models and implementations

sintel-dev/orion	2 papers	715 ★
------------------	----------	-------

Datasets

ODDS	FedTADBench
------	-------------

Most implemented papers

Most implemented Social Latest No code

Search for a paper, author or keyword

LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection

🔗 chickenbestlover/RNN-Time-series-Anomaly-Detection • 🍷 PyTorch • 1 Jul 2016

Mechanical devices such as engines, vehicles, aircrafts, etc., are typically instrumented with numerous sensors to capture the behavior and health of the machine.

🔗 6

[Paper](#)

[Code](#)

- Time Series Anomaly Detection
- Unsupervised Learning

LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, Gautam Shroff

{MALHOTRA.PANKAJ, ANUSHA.RAMAKRISHNAN, GAURANGI.ANAND, LOVEKESH.VIG, PUNEET.A, GAUTAM.SHROFF}@TCS.COM

TCS Research, New Delhi, India

Abstract

Mechanical devices such as engines, vehicles, aircrafts, etc., are typically instrumented with numerous sensors to capture the behavior and health of the machine. However, there are often external factors or variables which are not captured by sensors leading to time-series which

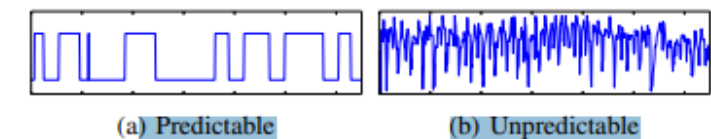


Figure 1. Readings for a manual control sensor.

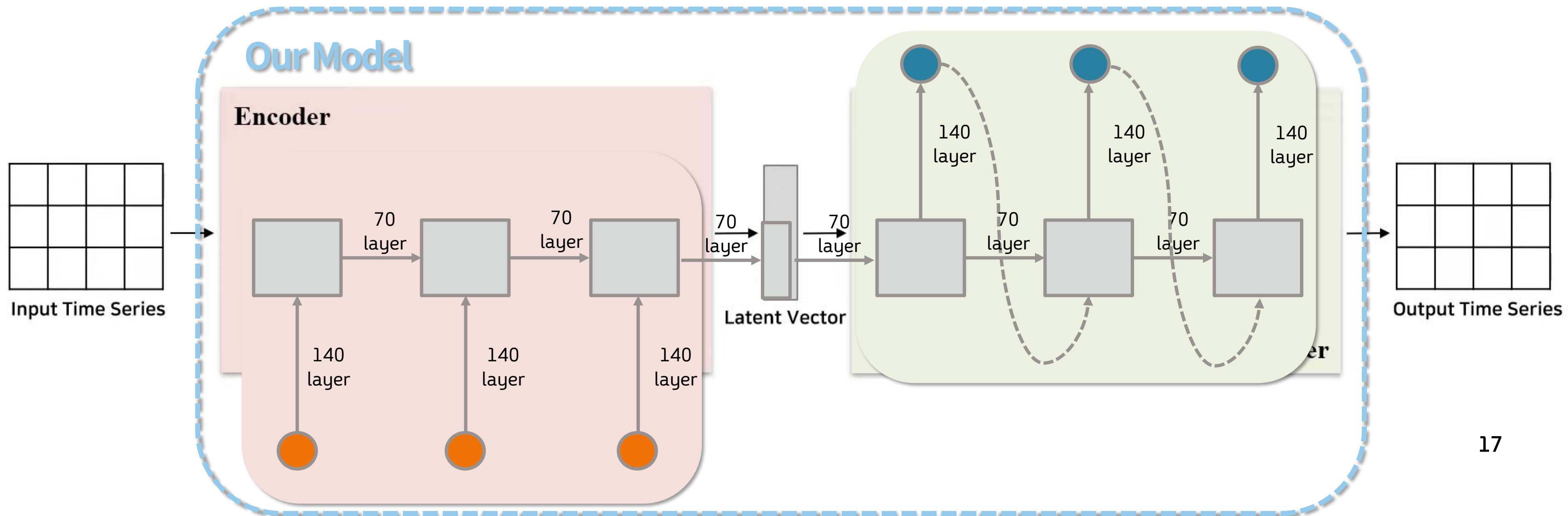
- RNN 기반의 시간 정보를 반영할 수 있는 모델
- LSTM + AutoEncoder 사용

- Reference : LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, Pankaj at el

LSTM-AE

모델링

- Sequence 데이터를 위한 LSTM 구조를 사용하는 AutoEncoder
- 시계열 데이터의 Anomaly Detection은 시간적 특성을 고려
- 이전 정보를 현재의 문제 해결에 활용할 수 있는 LSTM 네트워크 활용

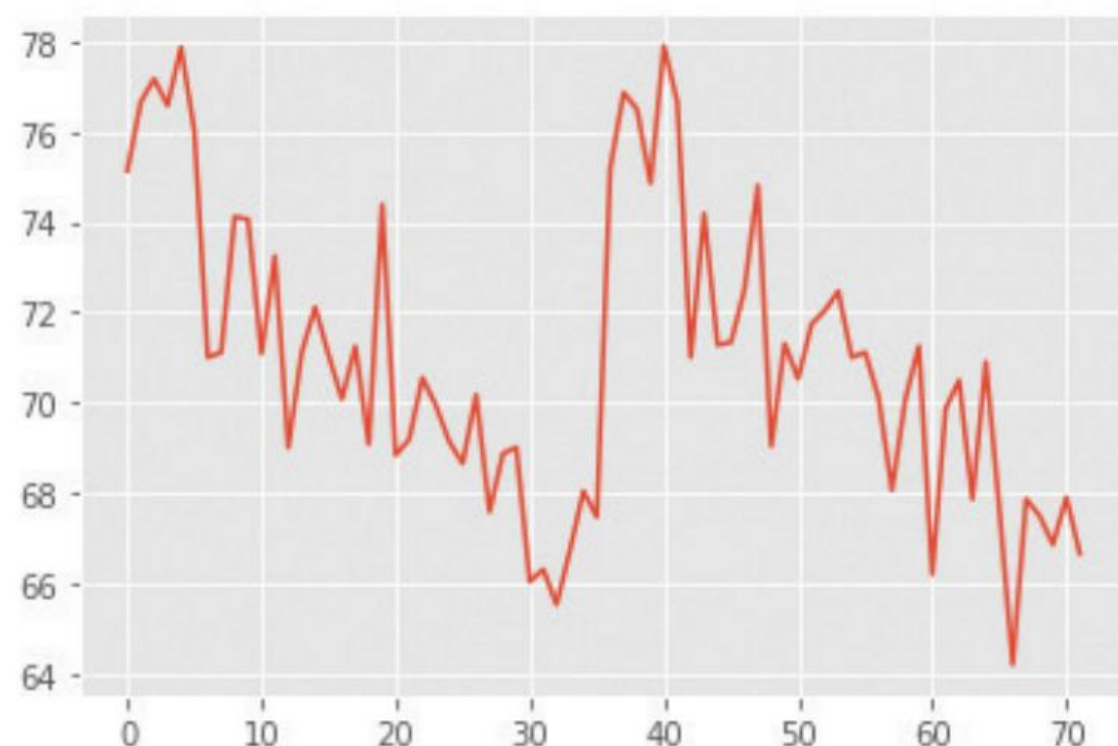


LSTM-AE

모델링

Model Parameters

- ★ Epoch : 130
- ★ LSTM Layer 개수 : 140개, 70개
- ★ Optimizer : Adam
- ★ Learning Rate : 0.001
- ★ Batch_size : 36 (3분 공정)
- ★ Loss Function : mse



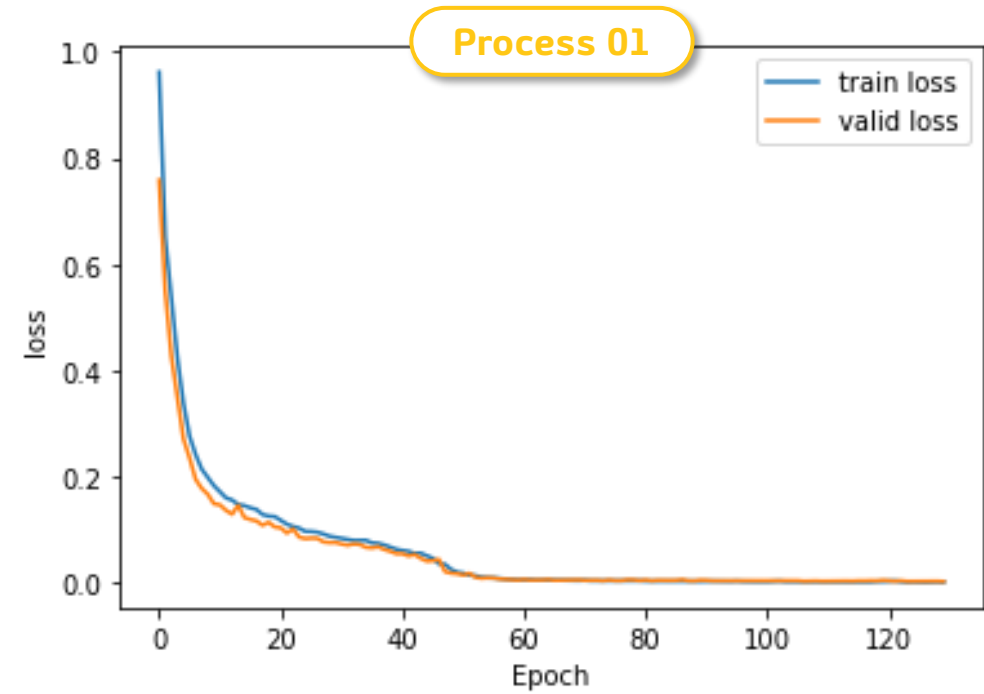
Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 3, 140)	80080
lstm_1 (LSTM)	(None, 70)	59080
repeat_vector (RepeatVector)	(None, 3, 70)	0
lstm_2 (LSTM)	(None, 3, 70)	39480
lstm_3 (LSTM)	(None, 3, 140)	118160
time_distributed (TimeDistr ibuted)	(None, 3, 2)	282

=====
Total params: 297,082
Trainable params: 297,082
Non-trainable params: 0
=====

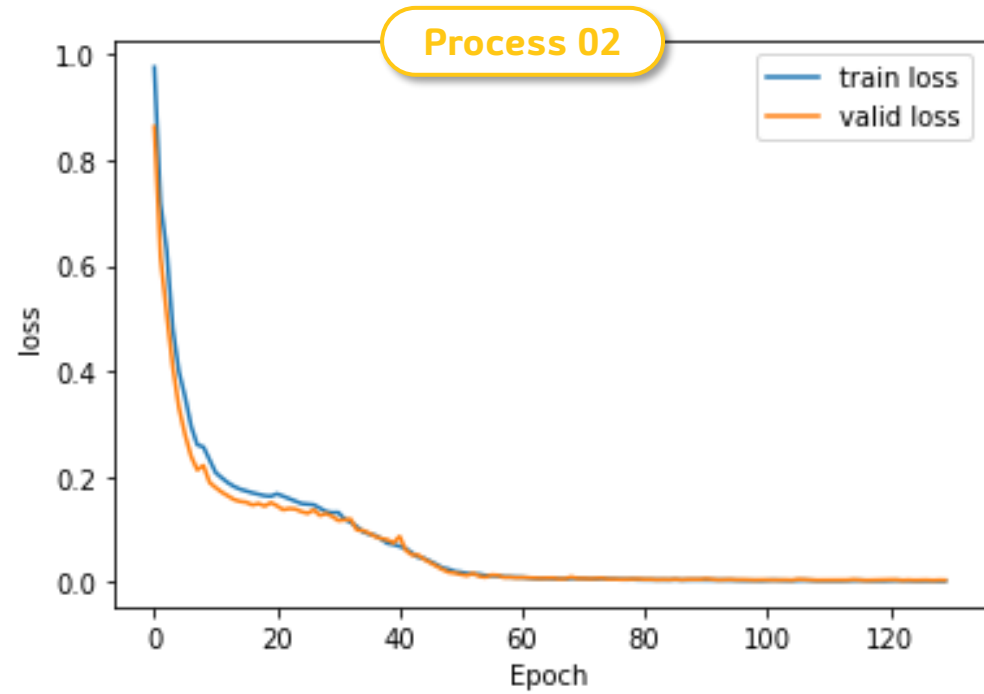
LSTM-AE

모델링



Threshold for this process is.. : 0.03253405361325141

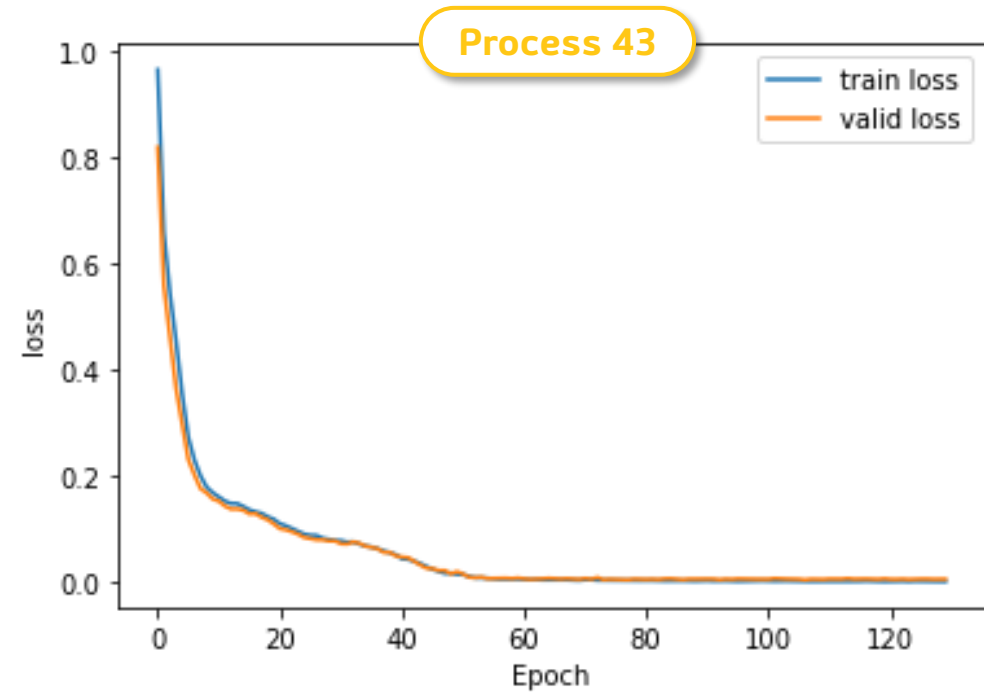
	Reconstruction error	True_class
count	249.000000	249.0
mean	0.002437	0.0
std	0.010828	0.0
min	0.000078	0.0
25%	0.000490	0.0
50%	0.000763	0.0
75%	0.001237	0.0
max	0.151656	0.0



Threshold for this process is.. : 0.07284140460001277

	Reconstruction error	True_class
count	213.000000	213.0
mean	0.003109	0.0
std	0.014040	0.0
min	0.000069	0.0
25%	0.000393	0.0
50%	0.000674	0.0
75%	0.001330	0.0
max	0.153434	0.0

...



Threshold for this process is.. : 0.08253452967017949

	Reconstruction error	True_class
count	213.000000	213.0
mean	0.006277	0.0
std	0.036301	0.0
min	0.000057	0.0
25%	0.000428	0.0
50%	0.000655	0.0
75%	0.001110	0.0
max	0.474949	0.0

...

LSTM-AE

모델링

```
result.set_index('index', inplace=True)

return model, result, label_sum
```

Process Inspection

```
In [*]: ▶ Log = pd.DataFrame()
diagnosis = ['Temp', 'Current']
for i in range(len(test)):
    print('In process proceeding...')
    event = test[i:i+1]
    Log = Log.append(event)
    time.sleep(4)
    display(Log)
    if len(Log) >= 3:
        model, diagnosis_result, label_sum = Inference(Log, i)
        model
        if label_sum >= 1:
            print('Anomaly detected!!')
            print('-----End Process-----\n\n')
            print('***Where an Anomaly Detected ?***')
            display(diagnosis_result)
            break
```

In process proceeding...

LSTM-AE

모델링

Anomaly detected!!

-----End Process-----

Where an Anomaly Detected ?

	Process	Temp	Current	NG
0	6	75.041983	1.55200	0
1	6	75.312474	1.49200	0
2	6	76.533664	1.69900	0
3	6	73.740424	1.70500	0
4	6	77.045185	1.73300	0
5	6	76.472491	1.67000	0
6	6	71.041983	1.74500	0
7	6	108.162219	1.12196	1



index	Label
0-2	normal
1-3	normal
2-4	normal
3-5	normal
4-6	normal
5-7	abnormal

LSTM-AE



센서 데이터에 의존하여 모델링을 한 결과 비교적 짧은 시간 내로 장비의 이상 유무를 확인할 수 있었습니다.
하지만 공장에서는 일분 일초가 매출과 관련된 상황에서 센서 데이터를 언제까지 기다리고 있을 수 없습니다.

따라서 **실시간으로 데이터를 분석**할 수 있는 **소리 데이터**를 활용해서 시스템을 구현하고자 합니다.

Isolation Forest

모델링

하지만 소리 데이터의 분석은 기존 센서의 수치형 데이터 분석과 차이가 존재합니다.



Sound_1



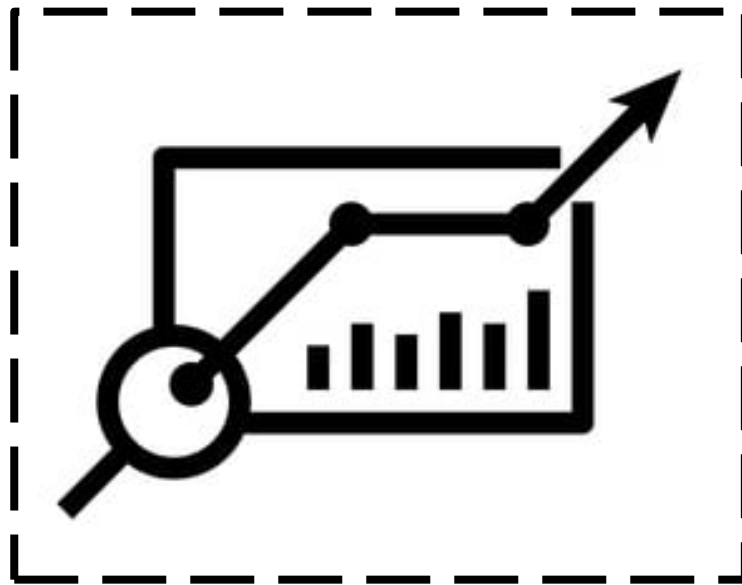
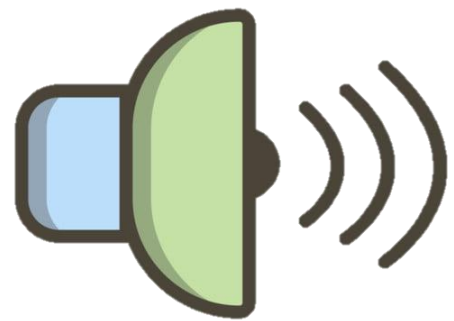
Sound_2

두 소리 중 어떤 소리가 장비에 이상이 있는 소리일까요?

또한 두 소리를 비교하기 위해서는 어떤 전처리가 필요할까요?

Isolation Forest

모델링

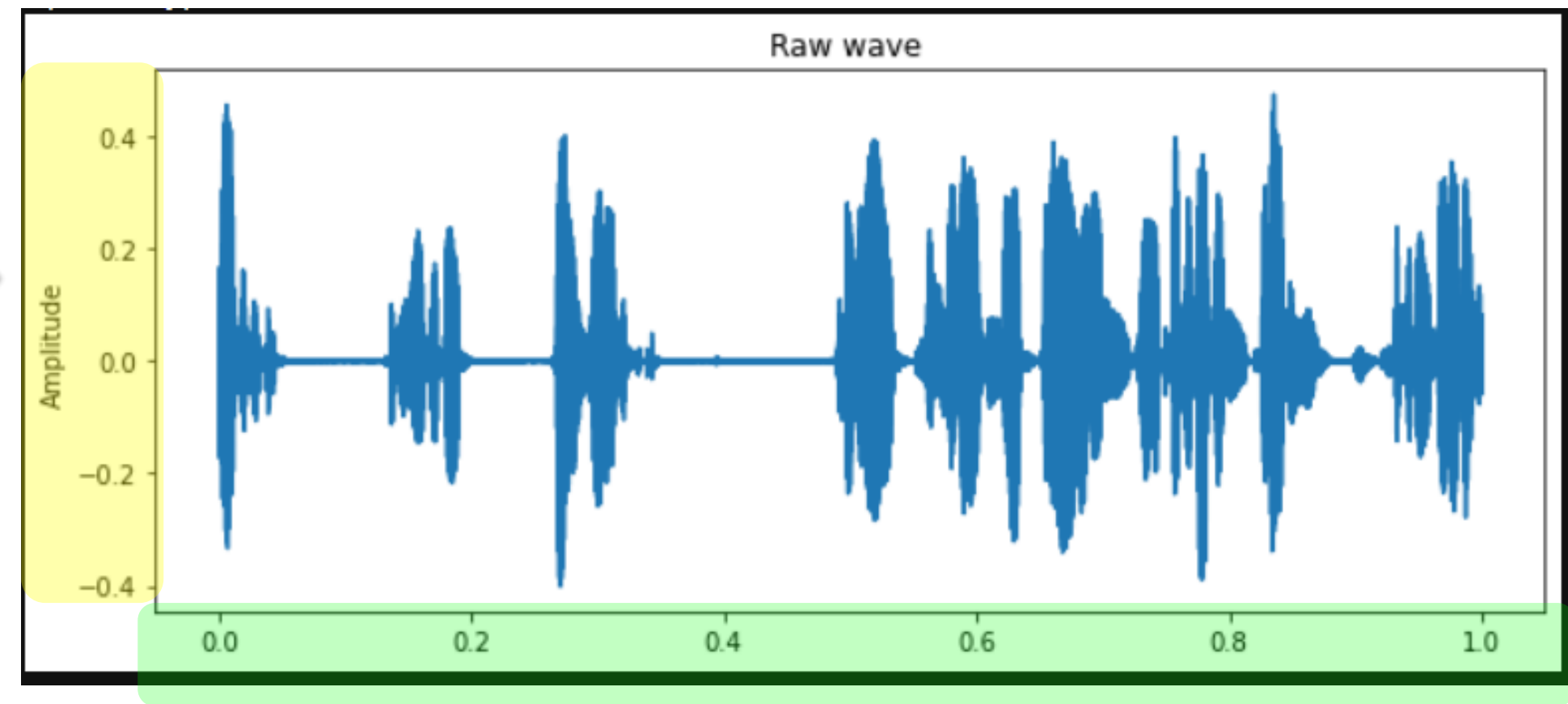


필요 라이브러리

```
pip install librosa  
pip install numpy  
pip install matplotlib
```



진폭 (Amplitude) Domain = 소리의 크기

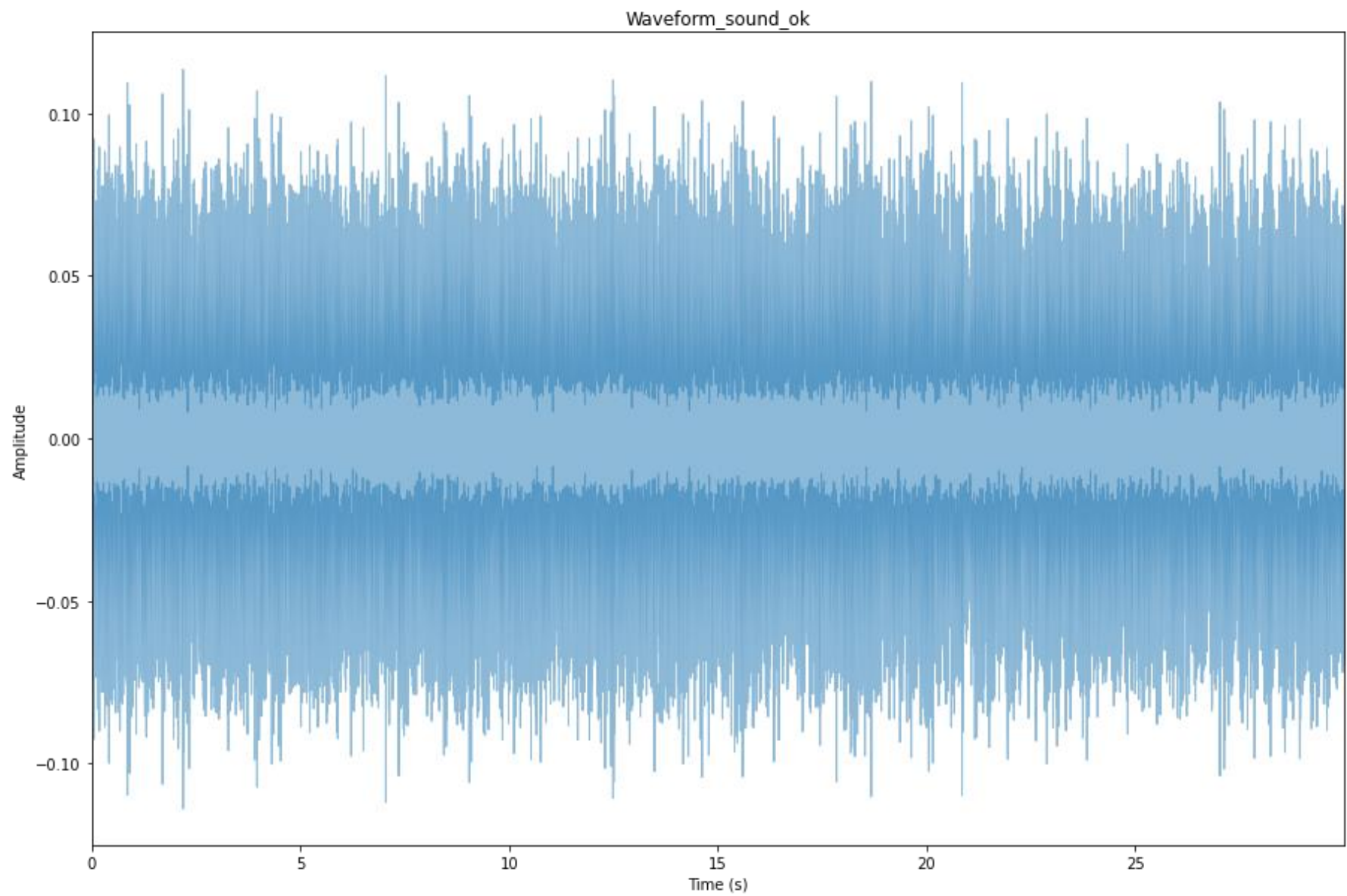


시간 (Time) Domain

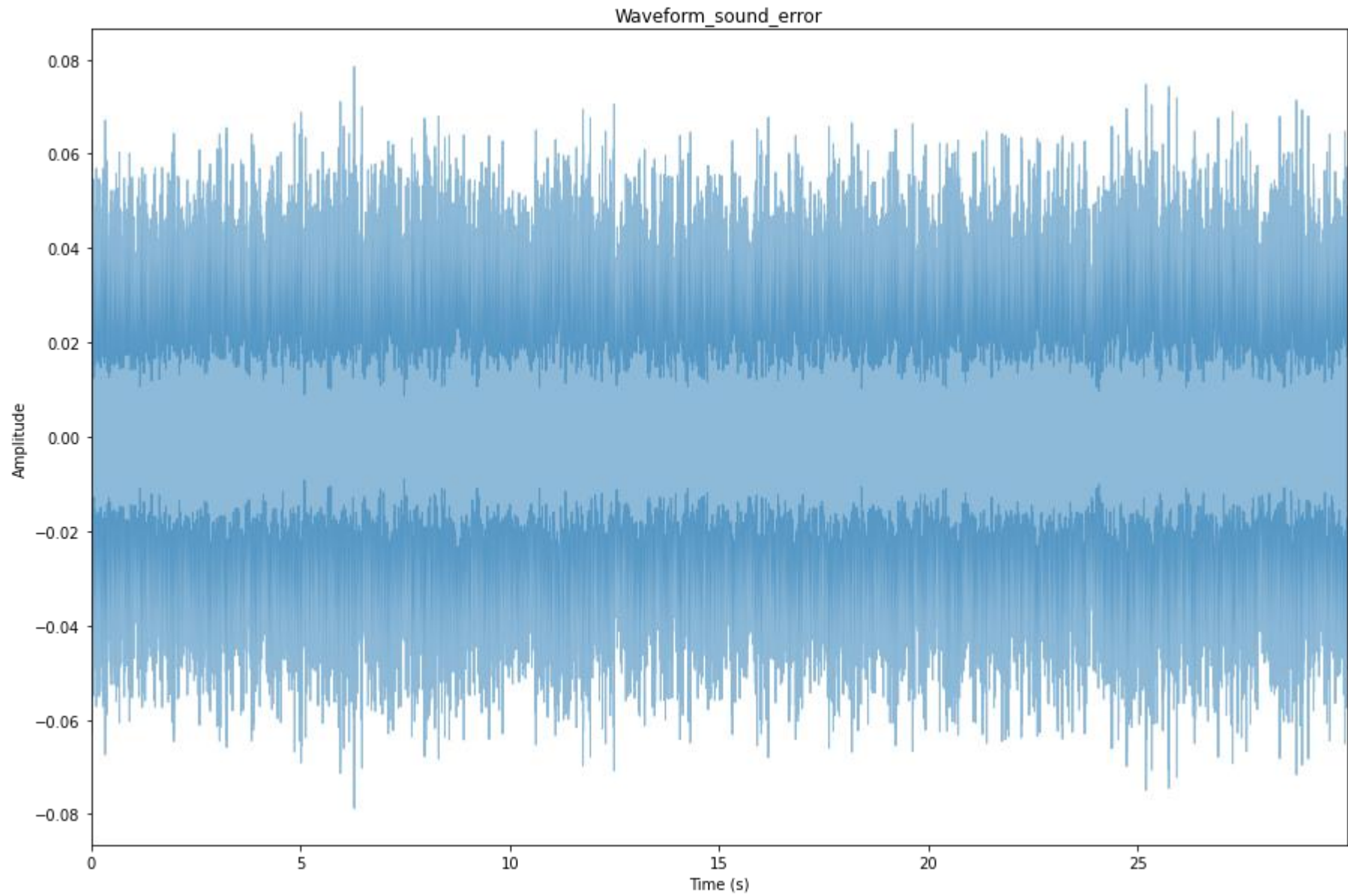


Isolation Forest

모델링



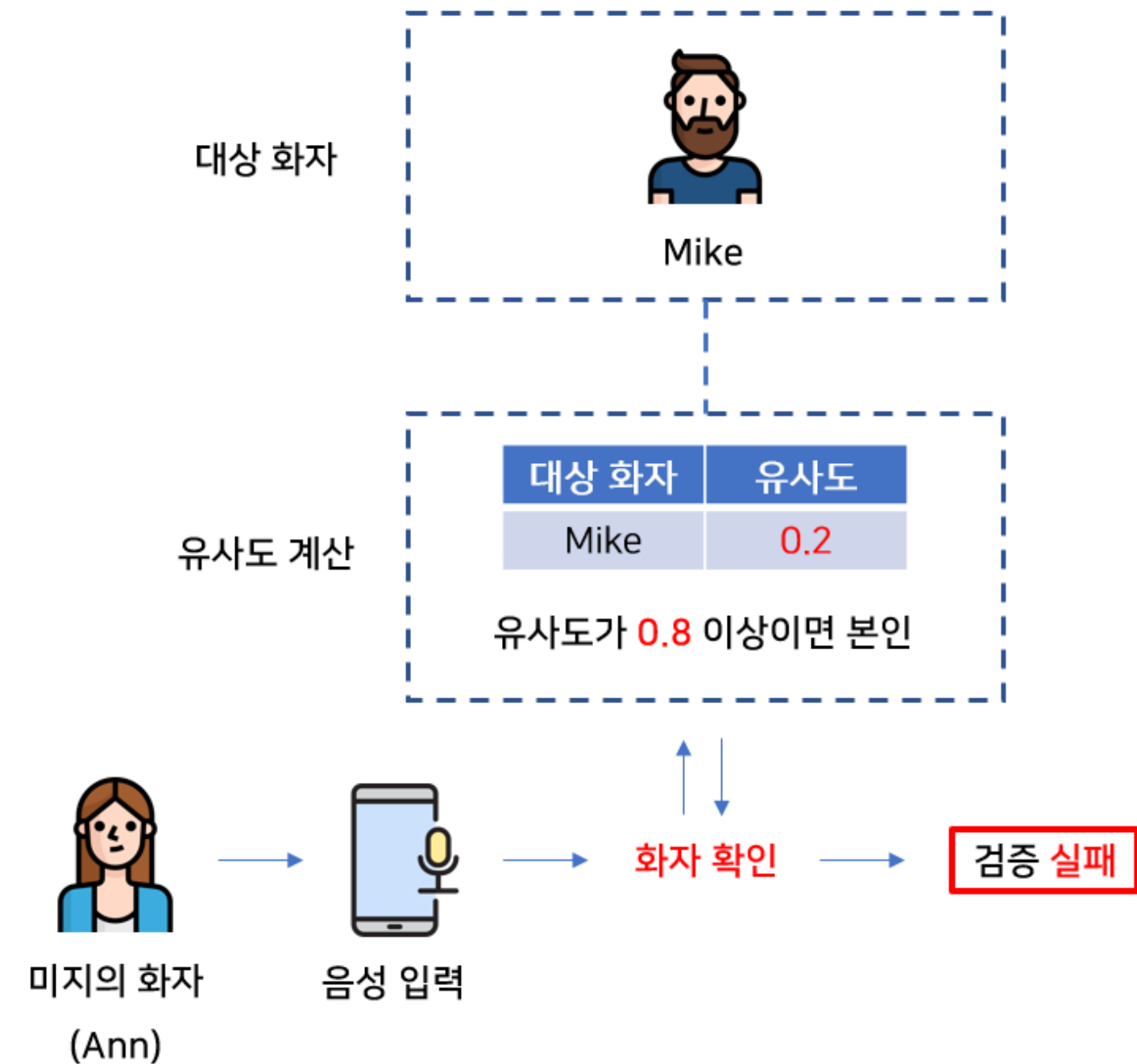
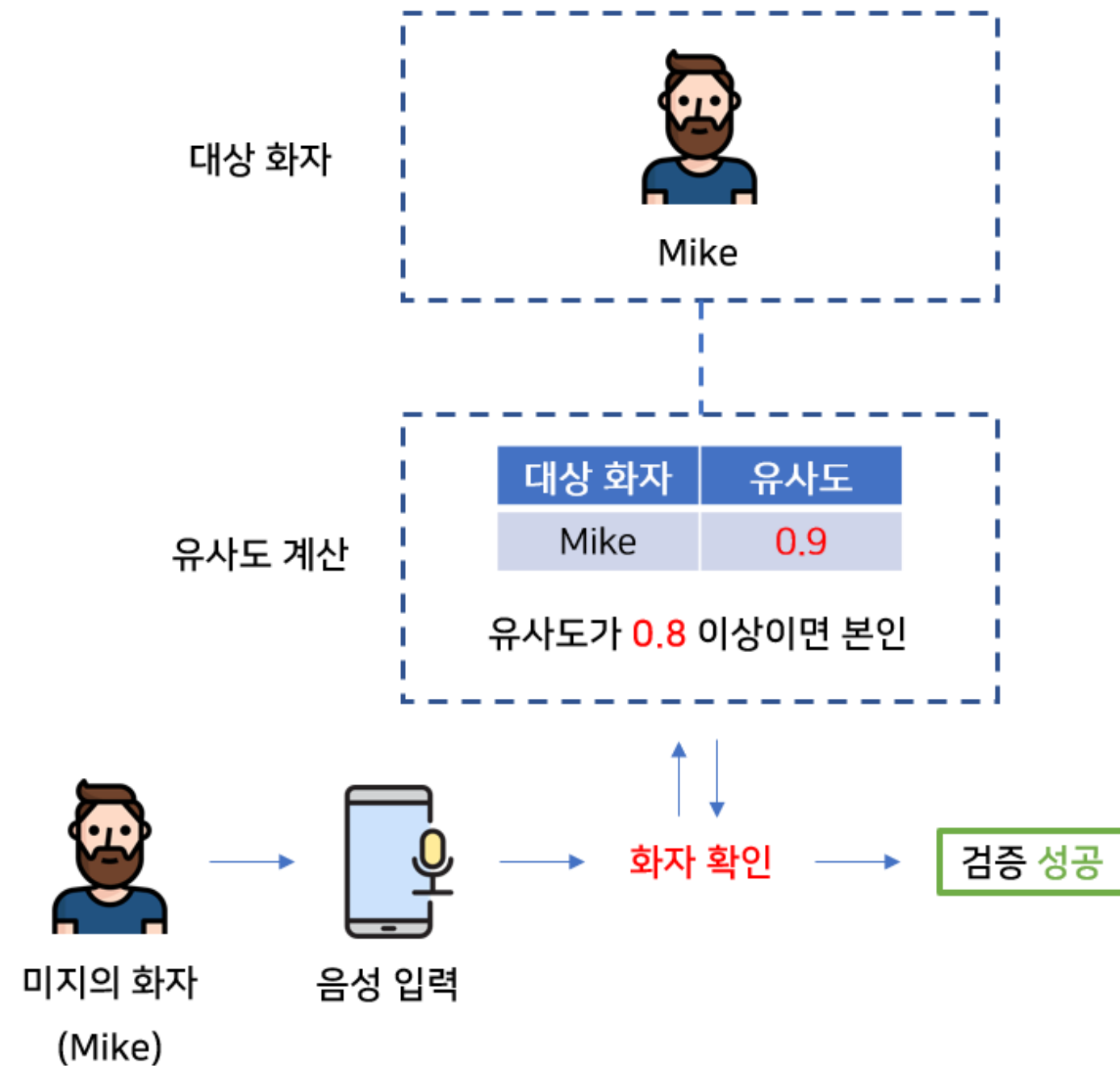
Sound_1 Waveform



Sound_2 Waveform

Isolation Forest

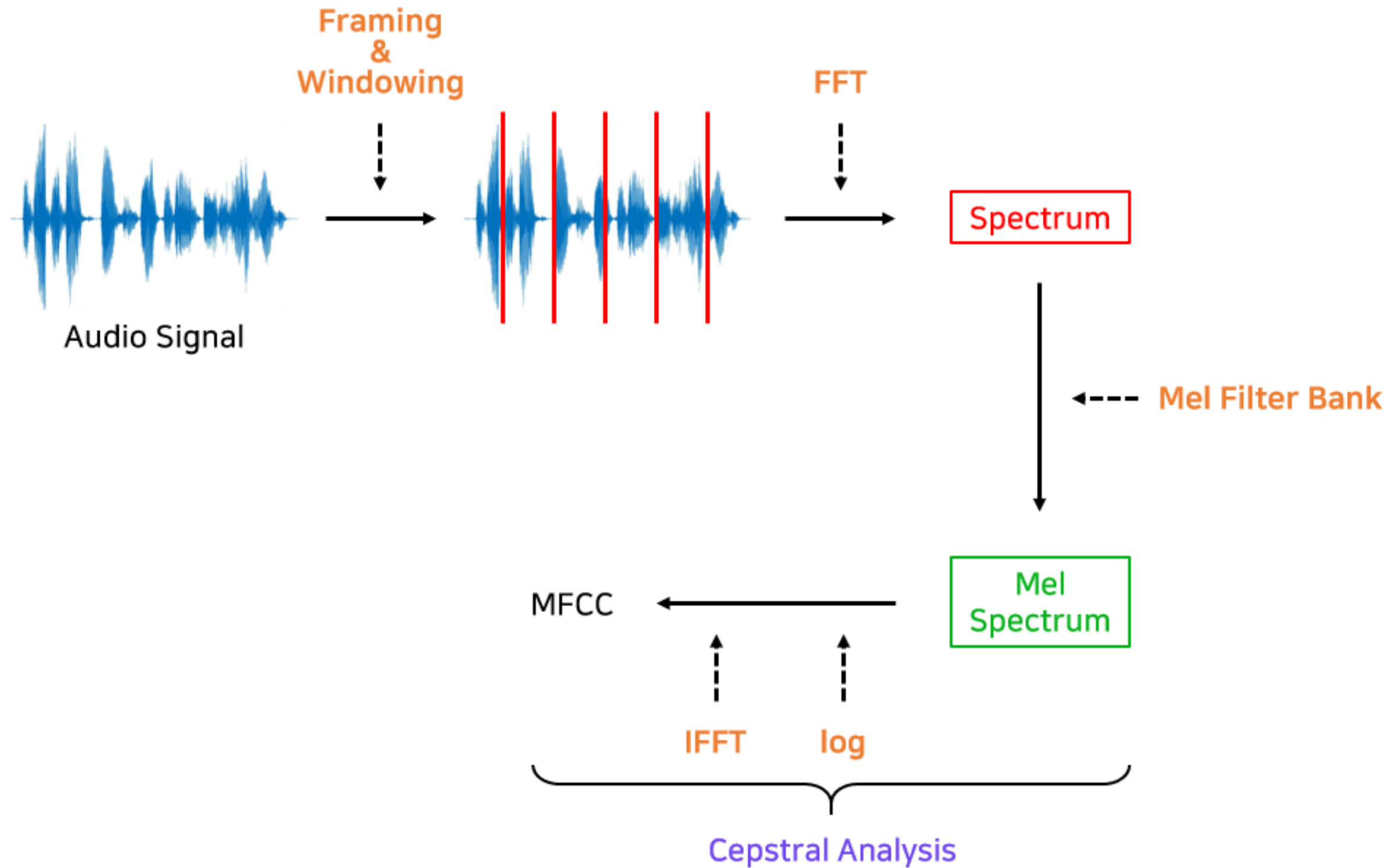
모델링



MFCC (Mel-Frequency Cepstral Coefficient)는 **소리의 고유한 특징**을 나타내는 수치!

Isolation Forest

모델링

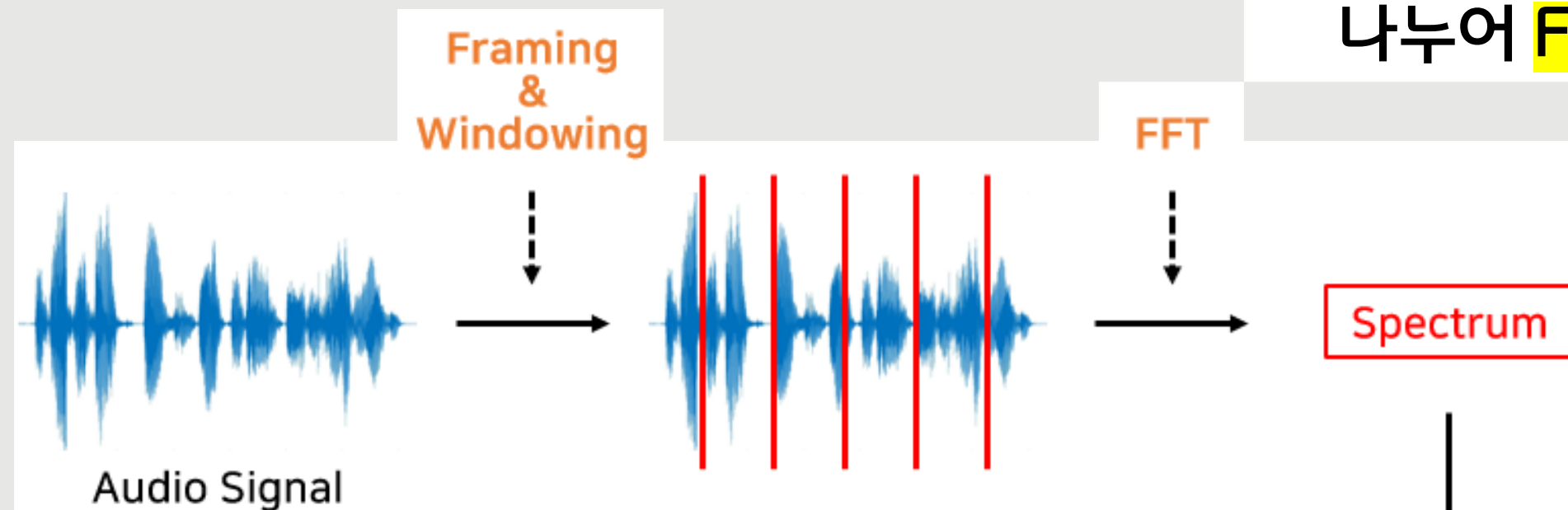


MFCC의 추출 과정

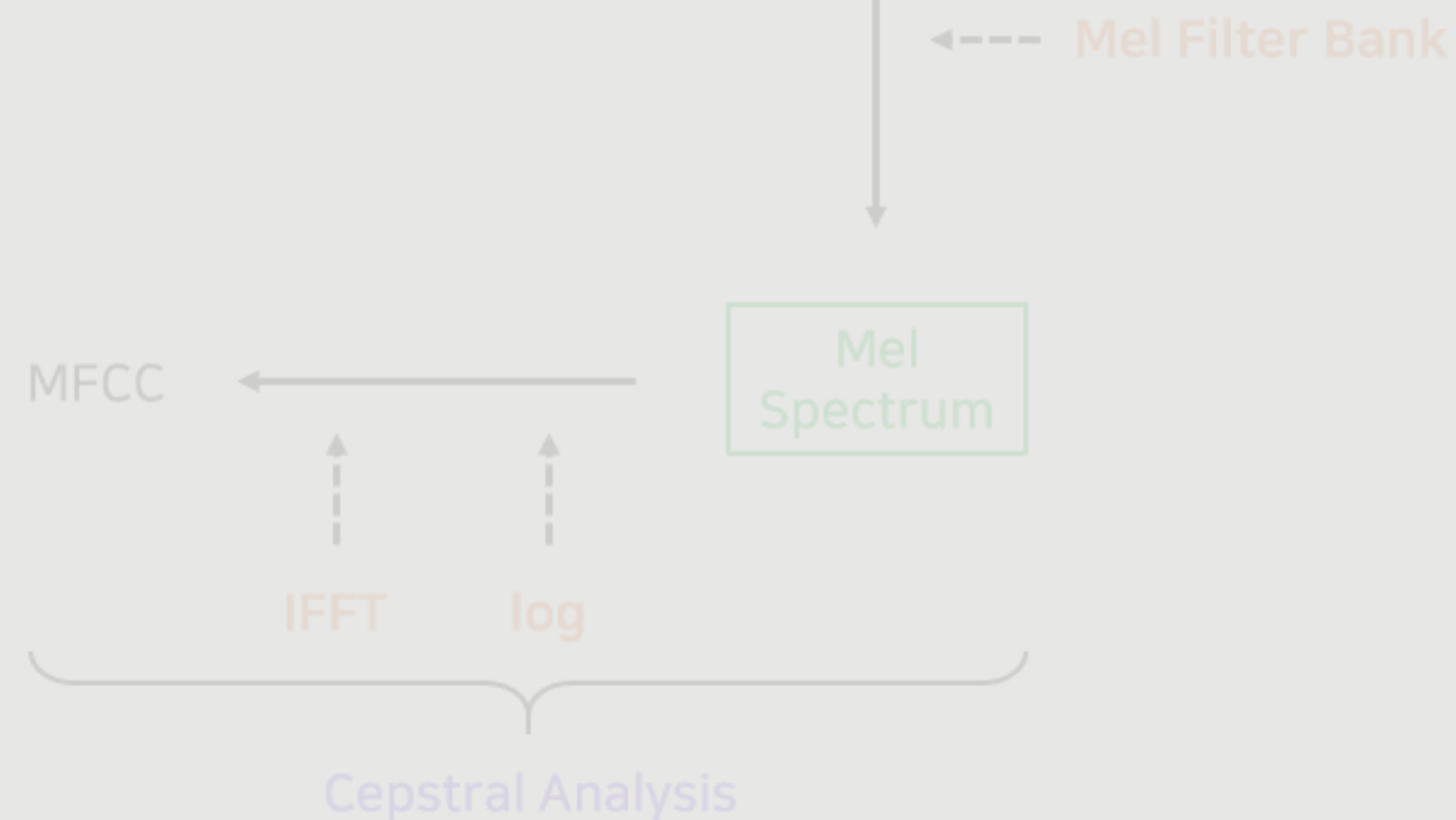
Isolation Forest

모델링

오디오 신호를 프레임별 (보통 20ms~ 40ms)로 나누어 **FFT**를 적용해 **Spectrum**을 구한다.



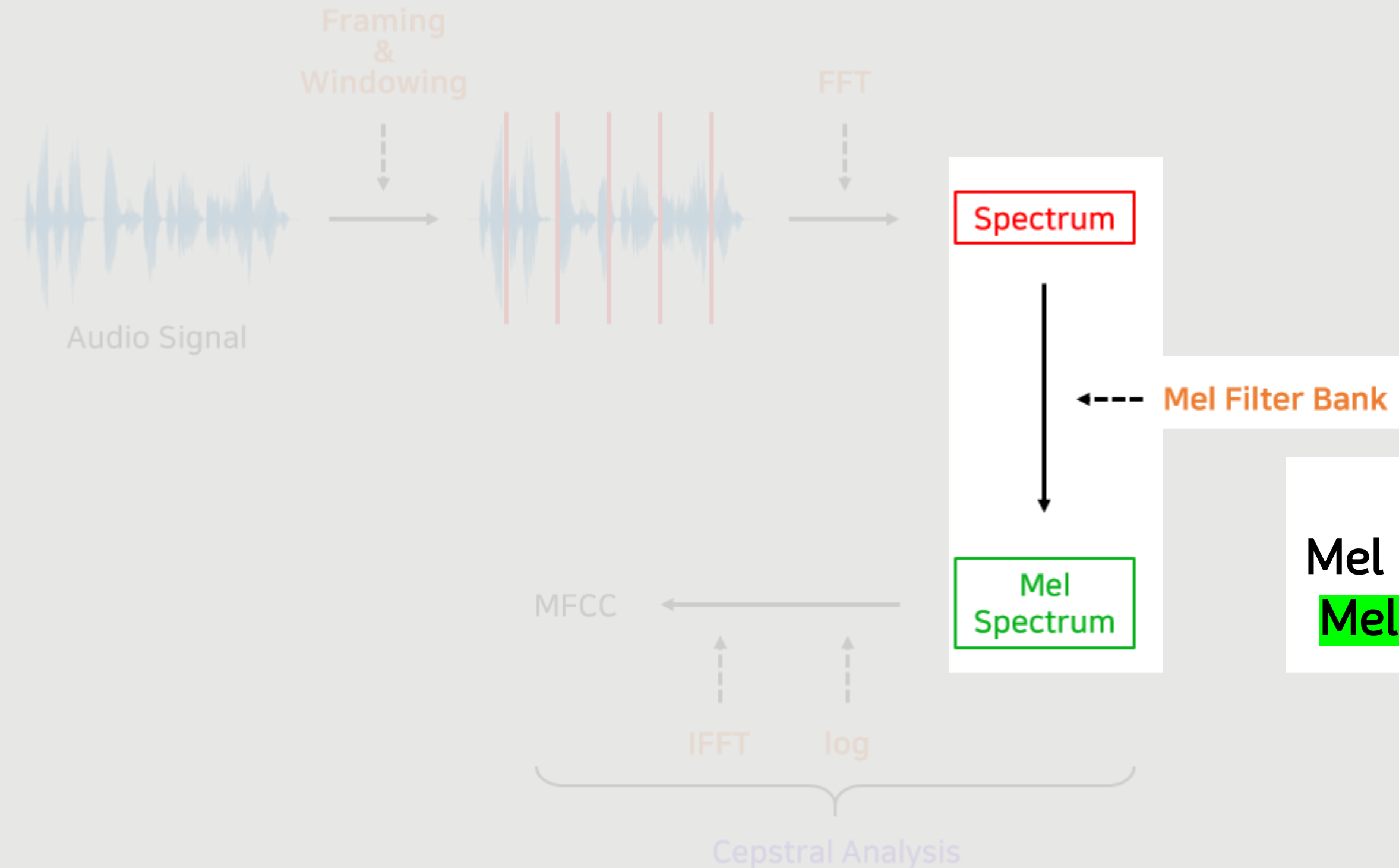
FFT (Fast Fourier Transform)
신호를 주파수 성분으로 변환하는 알고리즘
x축: 주파수, y축: 파워



MFCC의 추출 과정

Isolation Forest

모델링

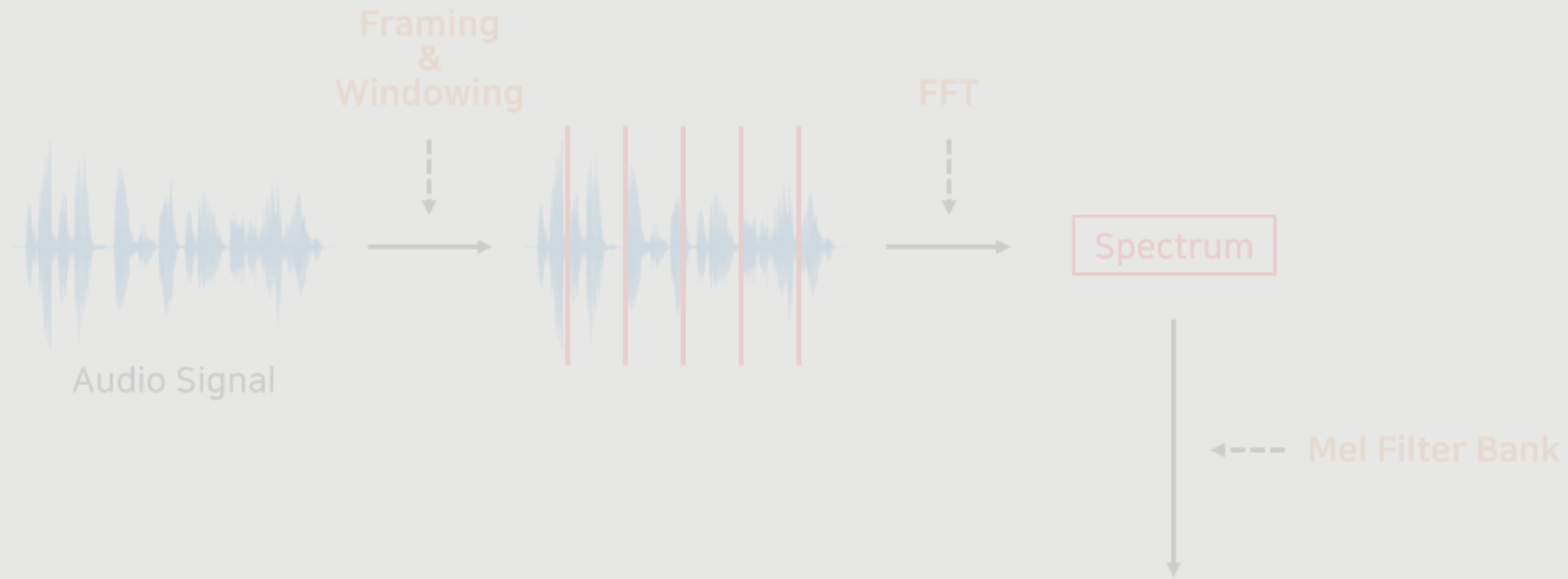


Spectrum에
Mel Filter Bank를 적용해
Mel Spectrum을 구한다

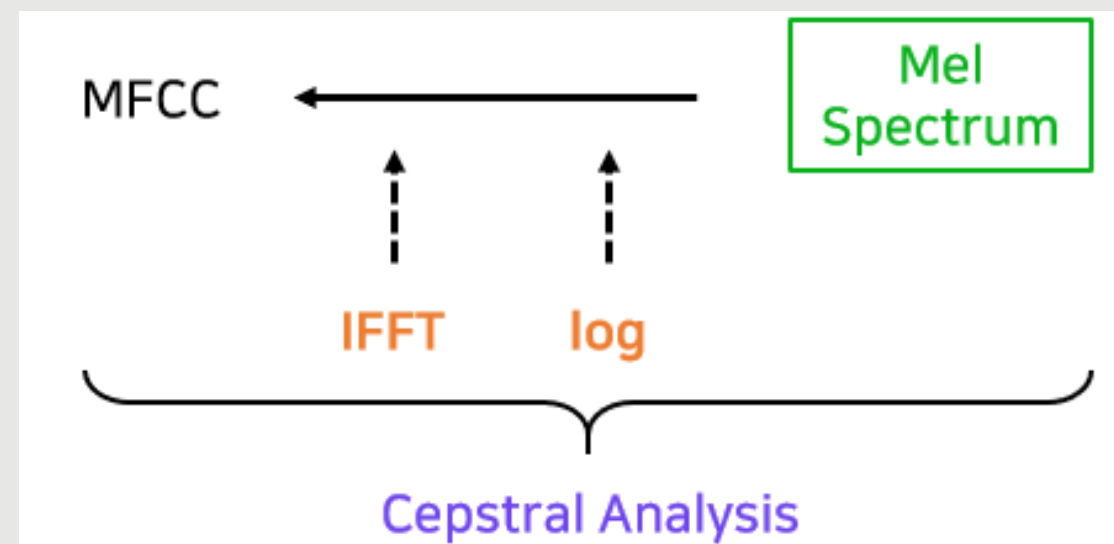
MFCC의 추출 과정

Isolation Forest

모델링



Mel Spectrum에 Cepstral 분석을 통해 MFCC를 구한다.

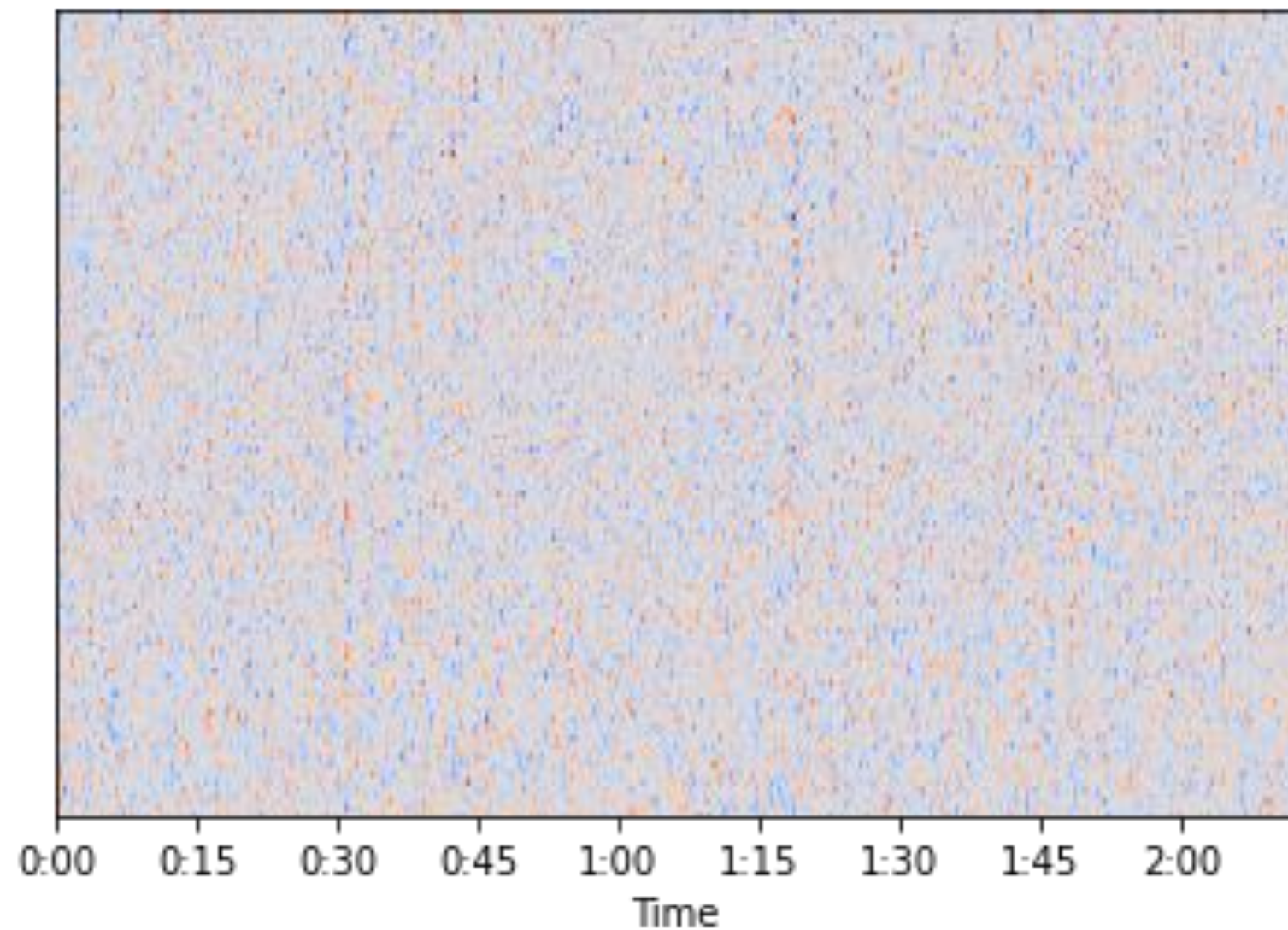


MFCC의 추출 과정

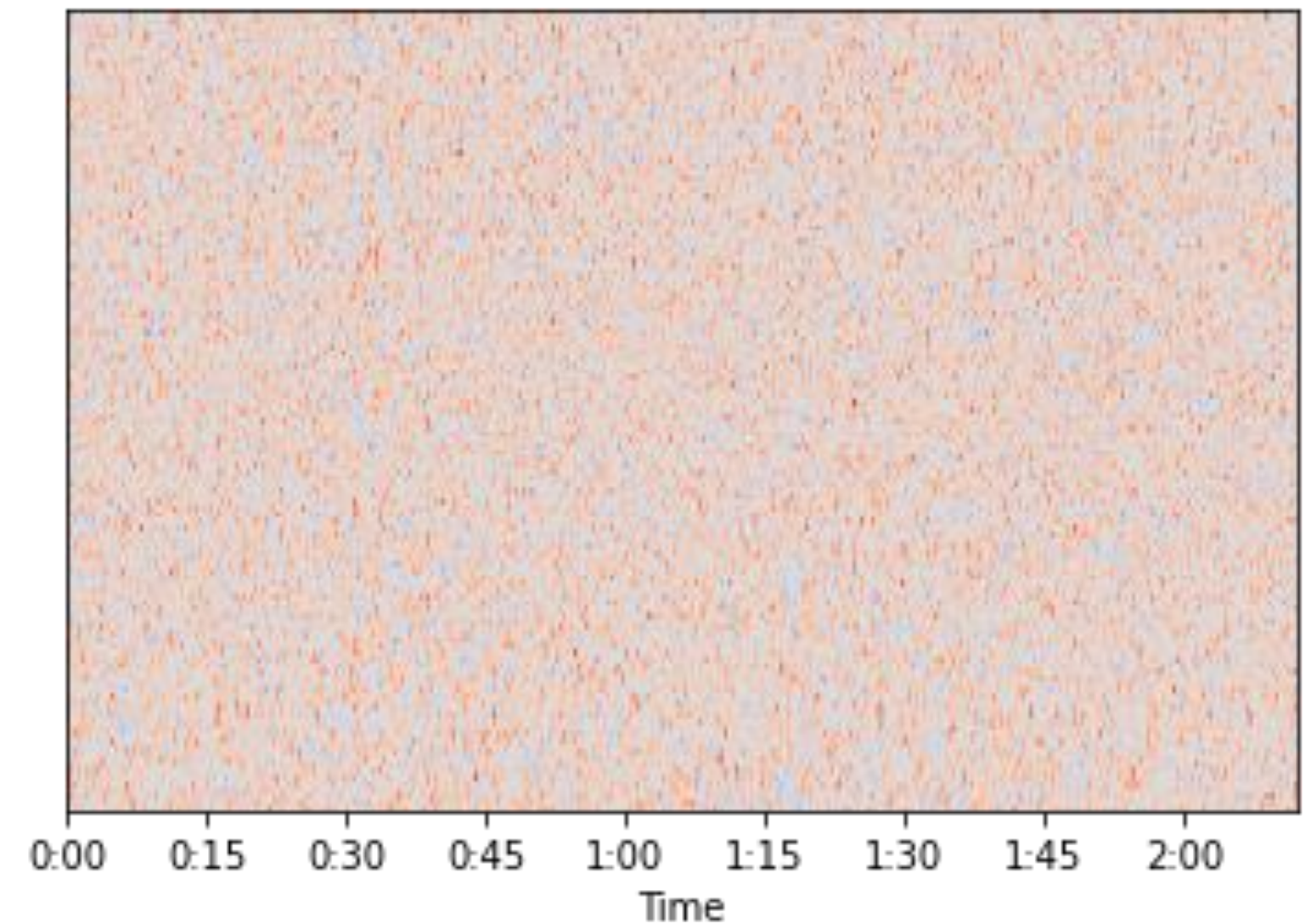
Isolation Forest

모델링

앞선 소리 데이터를 MFCC를 이용하면 아래와 같이 **소리의 고유한 특징**에 대한 데이터를 얻을 수 있습니다.



Sound_1_MFCC



Sound_2_MFCC

Isolation Forest

모델링

내 드라이브 > 중소기업_data > FAN_sound_OK		
이름 ↓	소유자	마지막으로 수정...
FAN_sound_157.wav	나	2021. 11. 23.
FAN_sound_156.wav	나	2021. 11. 23.
FAN_sound_155.wav	나	2021. 11. 23.
FAN_sound_154.wav	나	2021. 11. 23.
FAN_sound_153.wav	나	2021. 11. 23.
FAN_sound_152.wav	나	2021. 11. 23.
FAN_sound_151.wav	나	2021. 11. 23.
FAN_sound_150.wav	나	2021. 11. 23.
FAN_sound_149.wav	나	2021. 11. 23.
FAN_sound_148.wav	나	2021. 11. 23.

Sound_OK 데이터 170개

내 드라이브 > 중소기업_data > FAN_sound_error		
이름 ↓	소유자	마지막으로 수정...
FAN_sound_error_26.wav	나	2023. 2. 4.
FAN_sound_error_25.wav	나	2023. 2. 4.
FAN_sound_error_24.wav	나	2023. 2. 4.
FAN_sound_error_23.wav	나	2023. 2. 4.
FAN_sound_error_22.wav	나	2023. 2. 4.
FAN_sound_error_21.wav	나	2023. 2. 4.
FAN_sound_error_20.wav	나	2023. 2. 4.
FAN_sound_error_19.wav	나	2023. 2. 4.
FAN_sound_error_18.wav	나	2023. 2. 4.
FAN_sound_error_17.wav	나	2023. 2. 4.

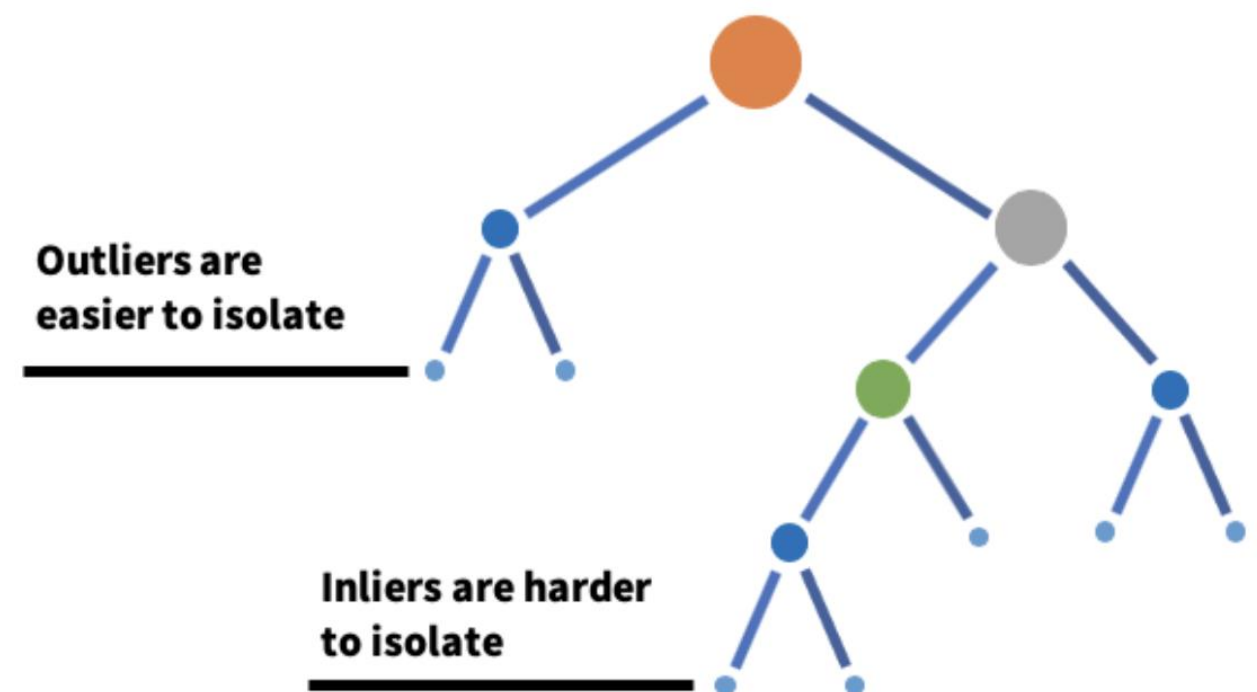
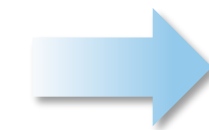
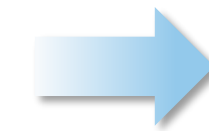
Sound_error 데이터 13개

```
1 def get_mfcc_feature_train(path):
2     features = []
3     for index in range(1,158):
4         y, sr = librosa.load(path+'FAN_sound_'+str(index)+'.wav', sr=CFG['SR'])
5
6         mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=CFG['N_MFCC'])
7
8         y_feature = []
9         for e in mfcc:
10             y_feature.append(np.max(e))
11         features.append(y_feature)
12     return features
```

MFCC feature

```
[25] 1 mfcc.shape
(128, 4135)
```

MFCC shape



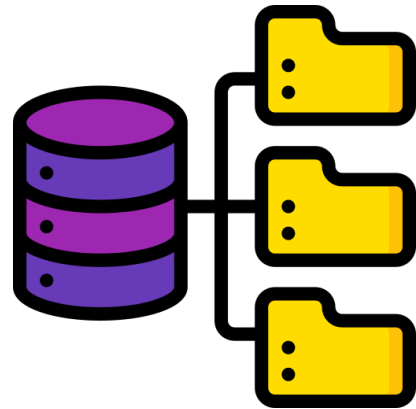
Isolation Forest

Isolation Forest

모델링

STEP 01

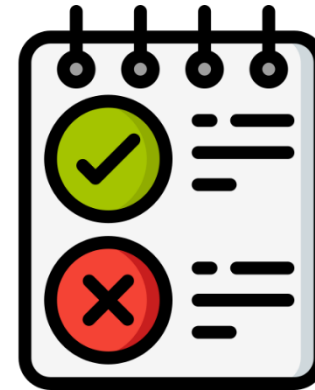
Train_set
정상 데이터



- 정상 데이터는 Train, Valid로 활용
- Train_set 157개 사용

STEP 02

Test_set
정상/비정상 데이터



- Error 데이터는 테스트 데이터로 활용
- NG 데이터와 1:1 비율로 정상 데이터도 추출하여 Test_set 구성

STEP 03

Parameter 튜닝



- n_estimator 최적값 설정 -> 6
- max_samples 최적값 설정-> 160

Isolation Forest


모델링

★ 재현율(Recall) 1을 목표로 모델링 진행

```
1 accuracy = metrics.accuracy_score(y_test, y_pred)
2 print("정확도:", accuracy)
3
4 precision = metrics.precision_score(y_test, y_pred)
5 print("정밀도:", precision)
6
7 ## 재현율 = TP / (TP + FN)=1 목표!
8 recall = metrics.recall_score(y_test, y_pred)
9 print("재현율:", recall)
10
11 f1 = metrics.f1_score(y_test, y_pred)
12 print("f1 점수:", f1)
```

정확도: 0.7307692307692307
정밀도: 0.65
재현율: 1.0
f1 점수: 0.7878787878787878

Confusion Matrix



TP 6	FN 7
FP 0	TN 13

05

효과 및 의의

01 기대효과

02 개선점 및 의의

기대효과

효과 및 의의



- ★ Process를 전부 진행하지 않고, 조기에 이상을 탐지하여 생산성 증대, 품질 향상, 시간/비용 절감 등의 부분에 기여할 수 있다.
- ★ 품질의 균일성을 확보하여 수율을 향상시킬 수 있다.
- ★ 1차적으로 소리 데이터를 활용한 단순 1차 검증을 통해 전체적인 공정 이상 탐지 시간을 단축할 수 있다.

개선점 및 의의

효과 및 의의

★ 일반화된 Threshold 설정 방식을 고안할 필요성이 있다.

★ LSTM AE의 Overfitting 문제를 해결하기 위해서 LSTM+CNN, GAN based Anomaly Detection 구현을 통해 모델링을 다양성을 고려해볼 수 있다.

Received November 21, 2018, accepted December 4, 2018, date of publication December 19, 2018, date of current version January 7, 2019.
Digital Object Identifier 10.1109/ACCESS.2018.2886457

DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series

MOHAMED ANDRÉ
¹Fachbereich
²German
Corresponding Author
This work was supported by the German Research Foundation (DFG) under the program SFB 1021/B1.

TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks

Alexander Geiger [*] <i>MIT</i> Cambridge, USA geigera@mit.edu	Dongyu Liu [*] <i>MIT</i> Cambridge, USA dongyu@mit.edu	Sarah Alnegheimish <i>MIT</i> Cambridge, USA smish@mit.edu
Alfredo Cuesta-Infante <i>Universidad Rey Juan Carlos</i> Madrid, Spain alfredo.cuesta@urjc.es	Kalyan Veeramachaneni <i>MIT</i> Cambridge, USA kalyanv@mit.edu	

USAD : UnSupervised Anomaly Detection on Multivariate Time Series

Julien Audibert julien.audibert@orange.com EURECOM Biot, France Orange Sophia Antipolis, France	Pietro Michiardi pietro.michiardi@eurecom.fr EURECOM Biot, France	Frédéric Guyard frederic.guyard@orange.com Orange Labs Sophia Antipolis, France
Sébastien Marti	Maria A. Zuluaga	

A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data

Chuxu Zhang^{§*}, Dongjin Song^{†*}, Yuncong Chen[†], Xinyang Feng^{†*}, Cristian Lumezanu[†], Wei Cheng[†], Jingchao Ni[†], Bo Zong[†], Haifeng Chen[†], Nitesh V. Chawla[§]

[§]University of Notre Dame, IN 46556, USA
[†]NEC Laboratories America, Inc., NJ 08540, USA
[‡]Columbia University, NY 10027, USA

[§]{czhang11,nchawla}@nd.edu, [†]{dsong,yuncong,lume,weicheng,jni,bzong,haifeng}@nec-labs.com, [‡]xf2143@columbia.edu



Thank You