




2023 경남 제조AI 해커톤 대회

AUTOENCODER를 통한 예지보전 자동화

PROVIBETECH





CONTENTS —

01 개발배경

02 프로젝트 진행

03 기대효과

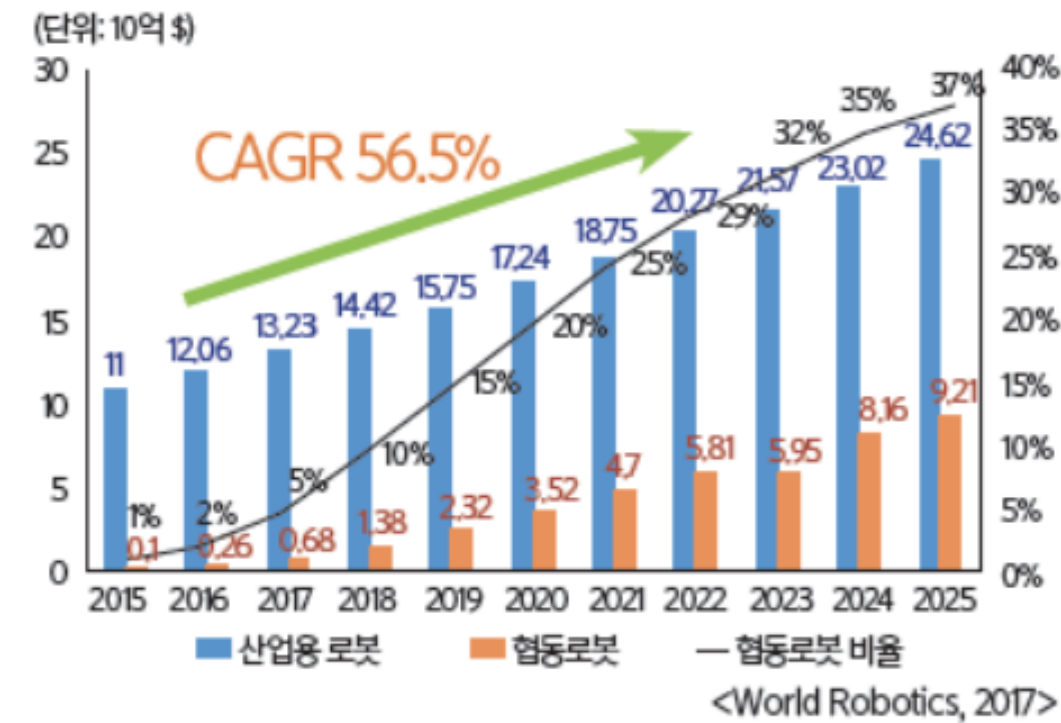


01

개발배경

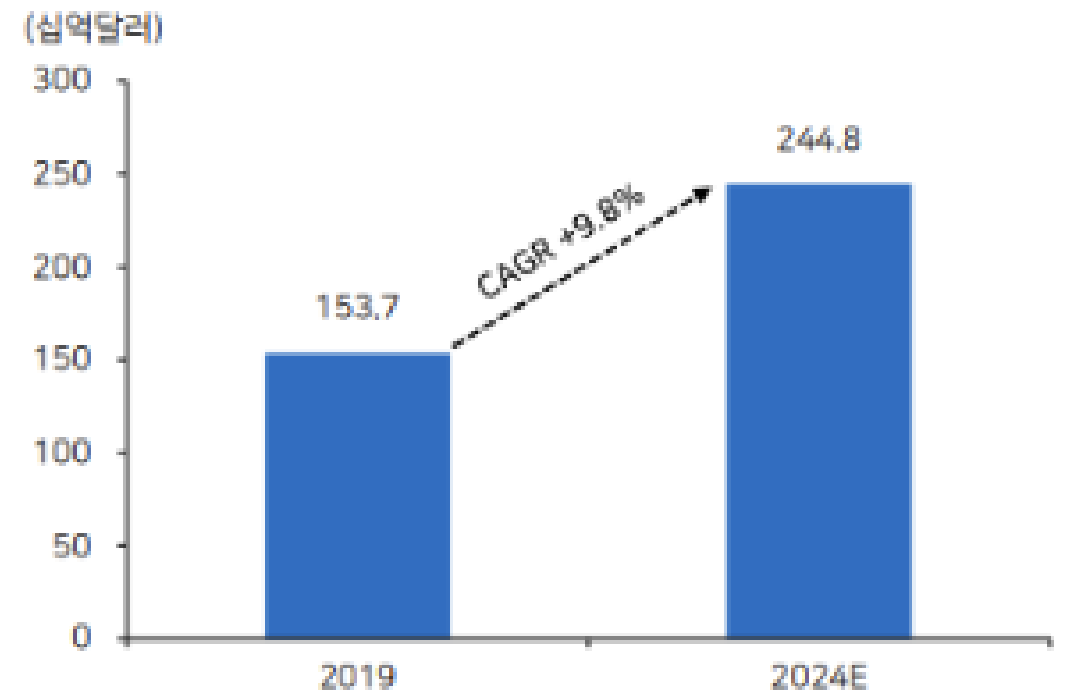
프로젝트 배경

그림 5 글로벌 산업용 로봇 시장과 협동로봇 시장 규모 전망



협업 로봇 시장의 성장

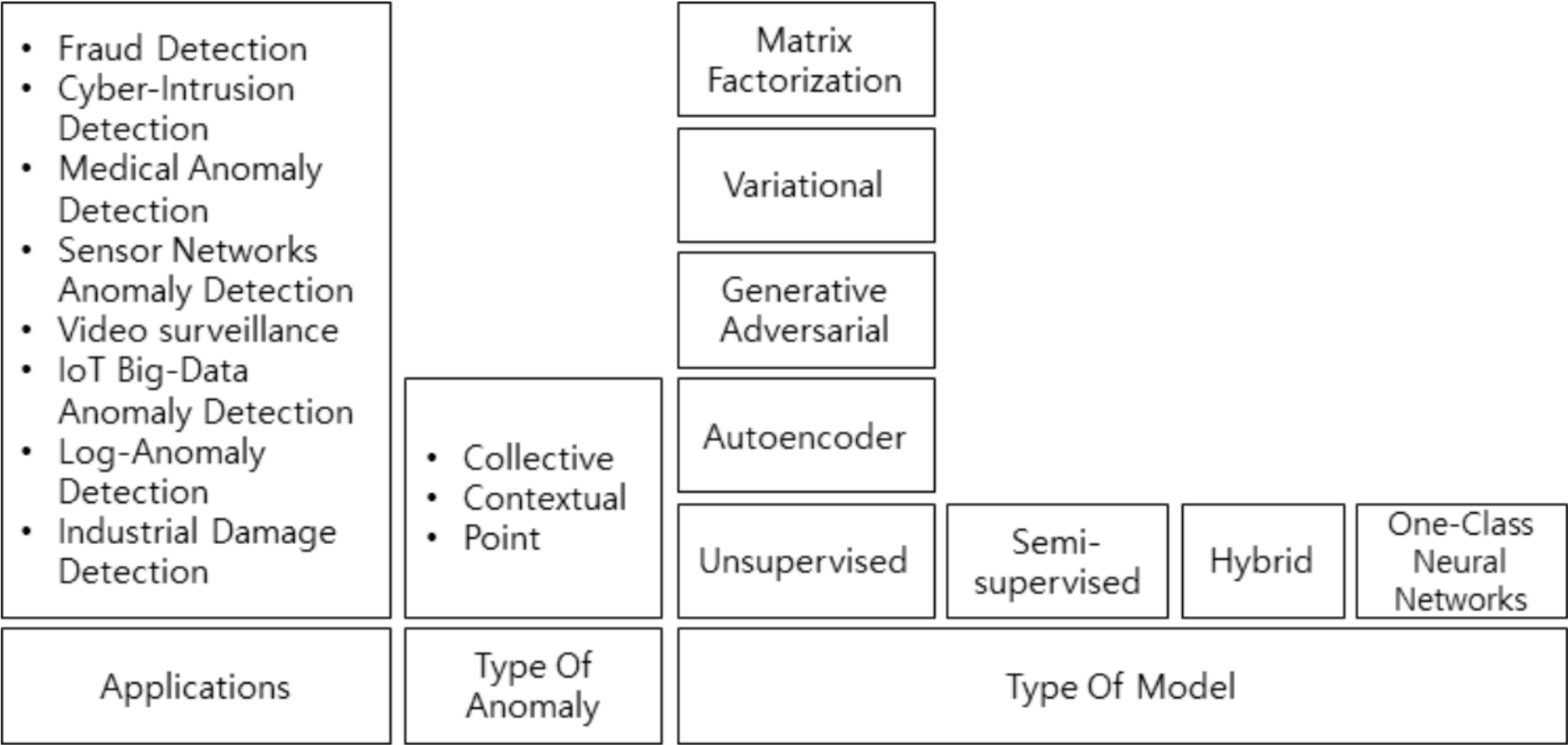
그림10 글로벌 스마트 팩토리 시장 규모 전망



자료: Markets&Markets, 메리츠증권 리서치센터

스마트팩토리 시장의 성장

공정 자동화 AI기술



AutoEncoder의 과거 사용 방식

- AutoEncoder를 통한 공정의 이상탐지

AutoEncoder → 사후해결 →

- AutoEncoder를 이용한 고장 진단

- AutoEncoder를 이용한 임계치 결정



프로젝트 목표

공정 부품 고장 예지
서비스

AutoEncoder



예지보전 자동화

이상 상황 탐지
서비스



— 02

프로젝트 진행

사용한 데이터

B
-0.03938
0.008532
0.024152
-0.00697
-0.01851
0.034415
0.032242
0.009029
0.025767
-0.02871
0.023772
0.071291
0.001669
-0.0429
0.013829

Normal
데이터

B
-0.01809
0.04418
0.028332
-0.00942
0.041995
0.003173
-0.07337
-0.03573
-0.03782
-0.03185
-0.02154
-0.05931
-0.03278
-0.03973
-0.074

Imbalance
데이터

B
-0.0029
-0.13482
0.049023
-0.09207
0.085629
0.146044
-0.10124
0.053184
-0.09296
-0.02224
0.120461
0.091202
-0.10405
-0.05601
0.209785

Ingression
데이터

데이터 주기

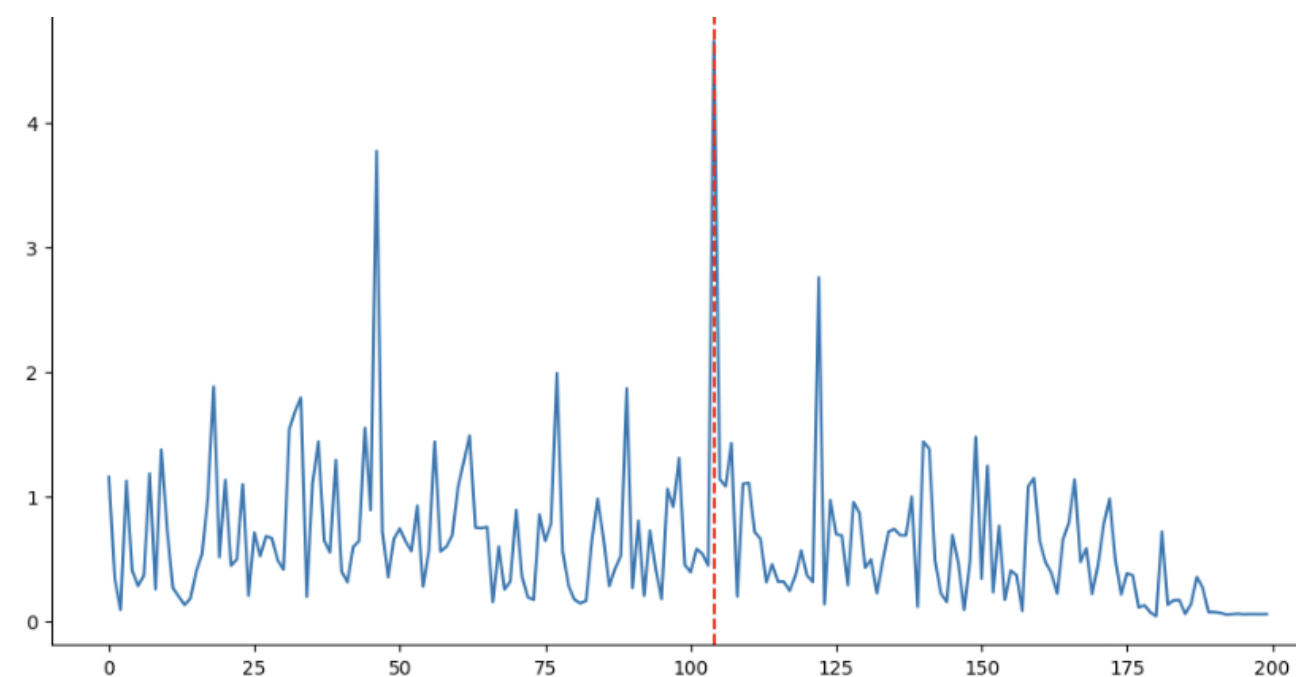
```
for frequency in tqdm(range(4, 300)):
    print(f'주기가 {frequency}에서')
    normal_features = split_and_extract_features(normal[:num_of_learning], frequency)
    impression_features = split_and_extract_features(impression[:num_of_learning], frequency)
    imbalance_features = split_and_extract_features(imbalance[:num_of_learning], frequency)

    X = pd.concat([normal_features, impression_features, imbalance_features])
    y = np.concatenate([np.zeros(len(normal_features)), np.ones(len(impression_features)), 2*np.ones(len(imbalance_features))])

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    clf = RandomForestClassifier()
    clf.fit(X_train, y_train)

    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
```



데이터 전처리 및 모델 생성

```
def ae_with_fft(df, frequency):
    extracted_features = []
    for i in range(0, len(df), frequency):
        segment = df.iloc[i:i+frequency]
        if len(segment) == frequency:
            # FFT 적용
            fft_values = np.fft.fft(segment)

            half_length = len(fft_values) // 2
            fft_values_half = fft_values[:half_length]

            magnitude = np.abs(fft_values_half)
            extracted_features.append(magnitude)
    return pd.DataFrame(extracted_features)

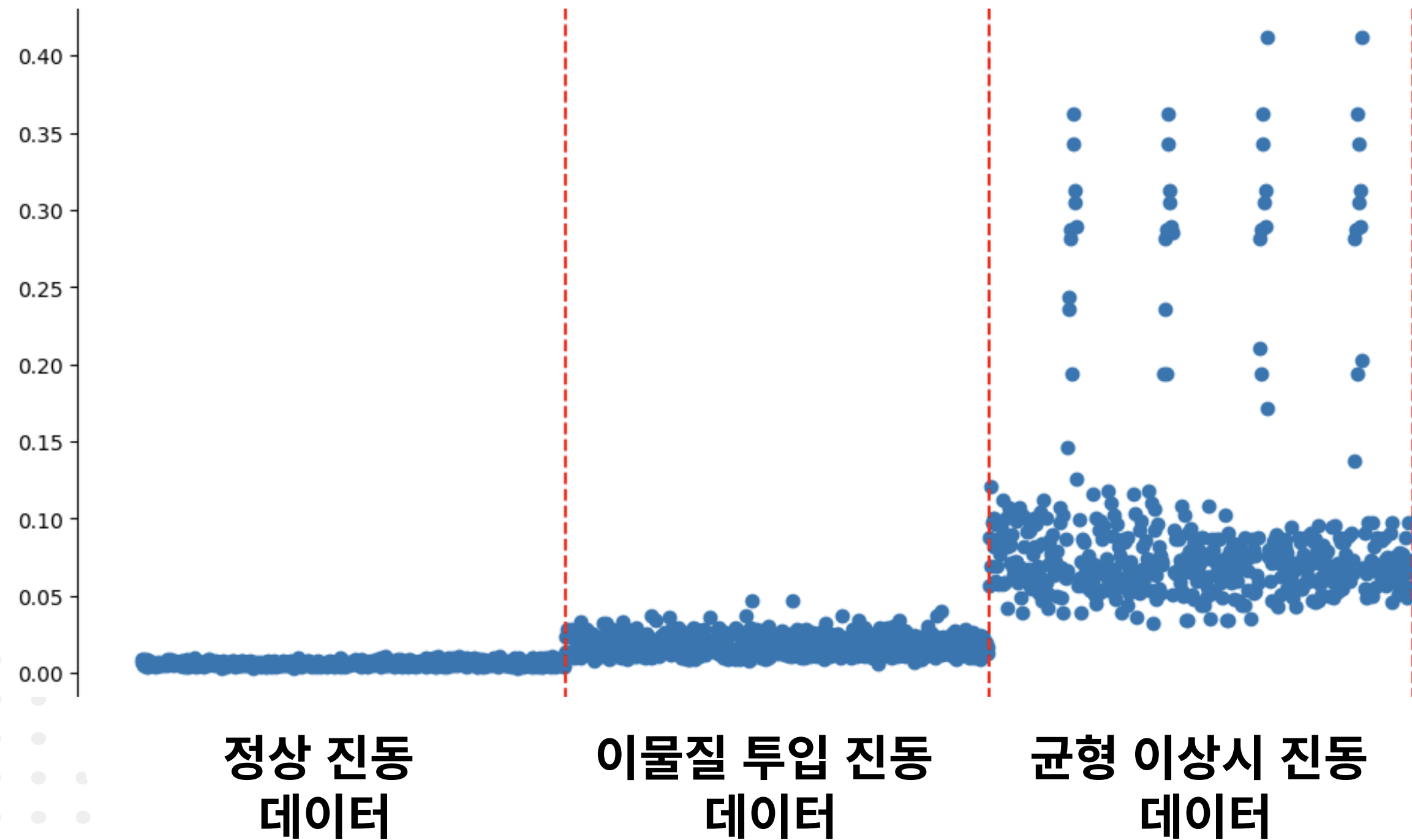
num_of_learning = 49979
frequency = 104

normal_ae = ae_with_fft(renormal[:num_of_learning], frequency)
normal_ae = np.array(normal_ae).reshape(-1,1,frequency)
print("normal 데이터 shape:",normal_ae.shape)
```

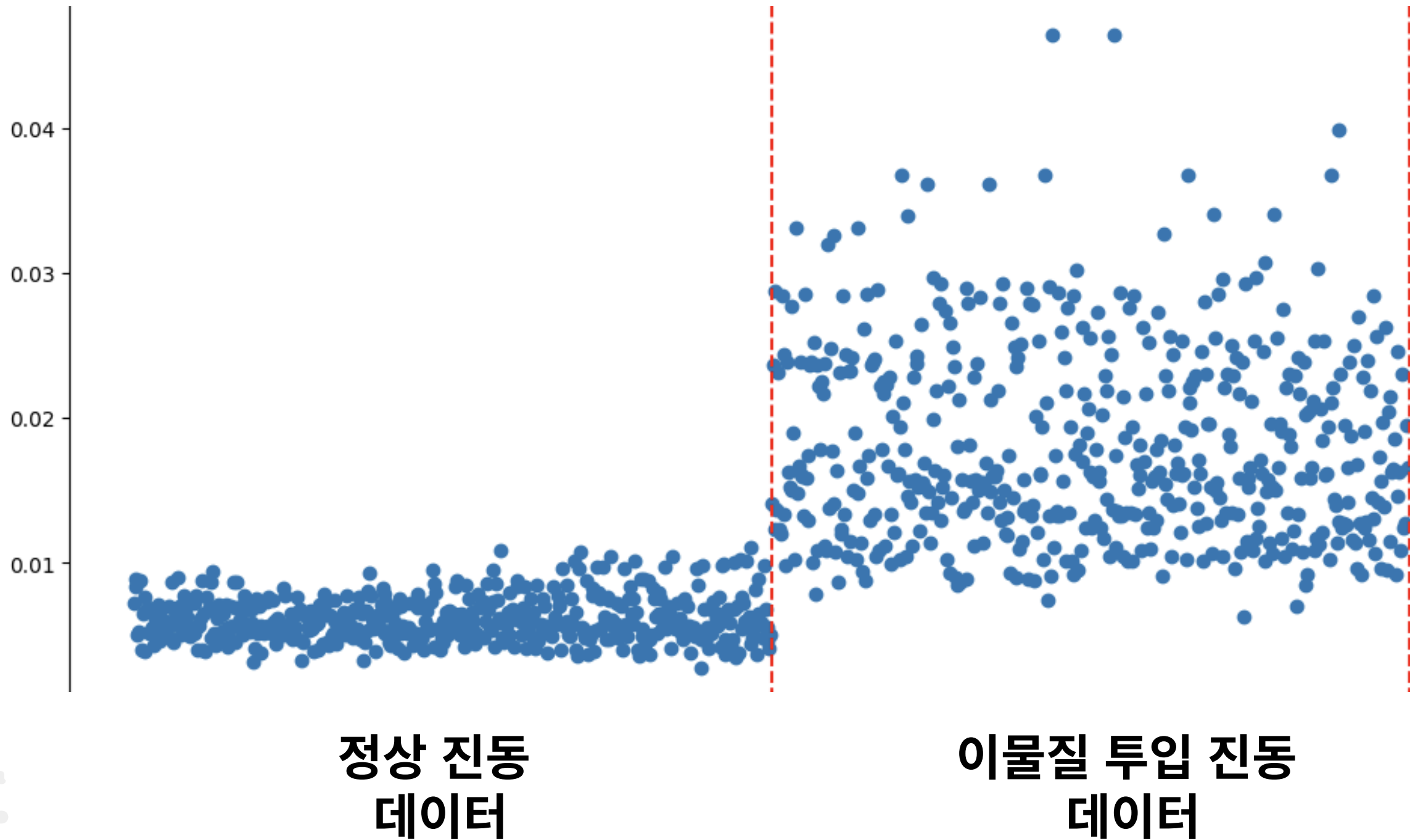
normal 데이터 shape: (240, 1, 104)

Layer (type)	Output Shape	Param #
=====		
conv1d (Conv1D)	(None, 1, 128)	213120
dense (Dense)	(None, 1, 32)	4128
lstm_1 (LSTM)	(None, 16)	3136
dense_1 (Dense)	(None, 16)	272
repeat_vector (RepeatVector)	(None, 1, 16)	0
dense_2 (Dense)	(None, 1, 16)	272
lstm_2 (LSTM)	(None, 1, 16)	2112
dense_3 (Dense)	(None, 1, 32)	544
conv1d_1 (Conv1D)	(None, 1, 128)	65664
time_distributed (TimeDistributed)	(None, 1, 104)	13416
...		
12/12 [=====] - 0s 3ms/step - loss: 0.0057 - val_loss: 0.0054		
Model Save Complete		

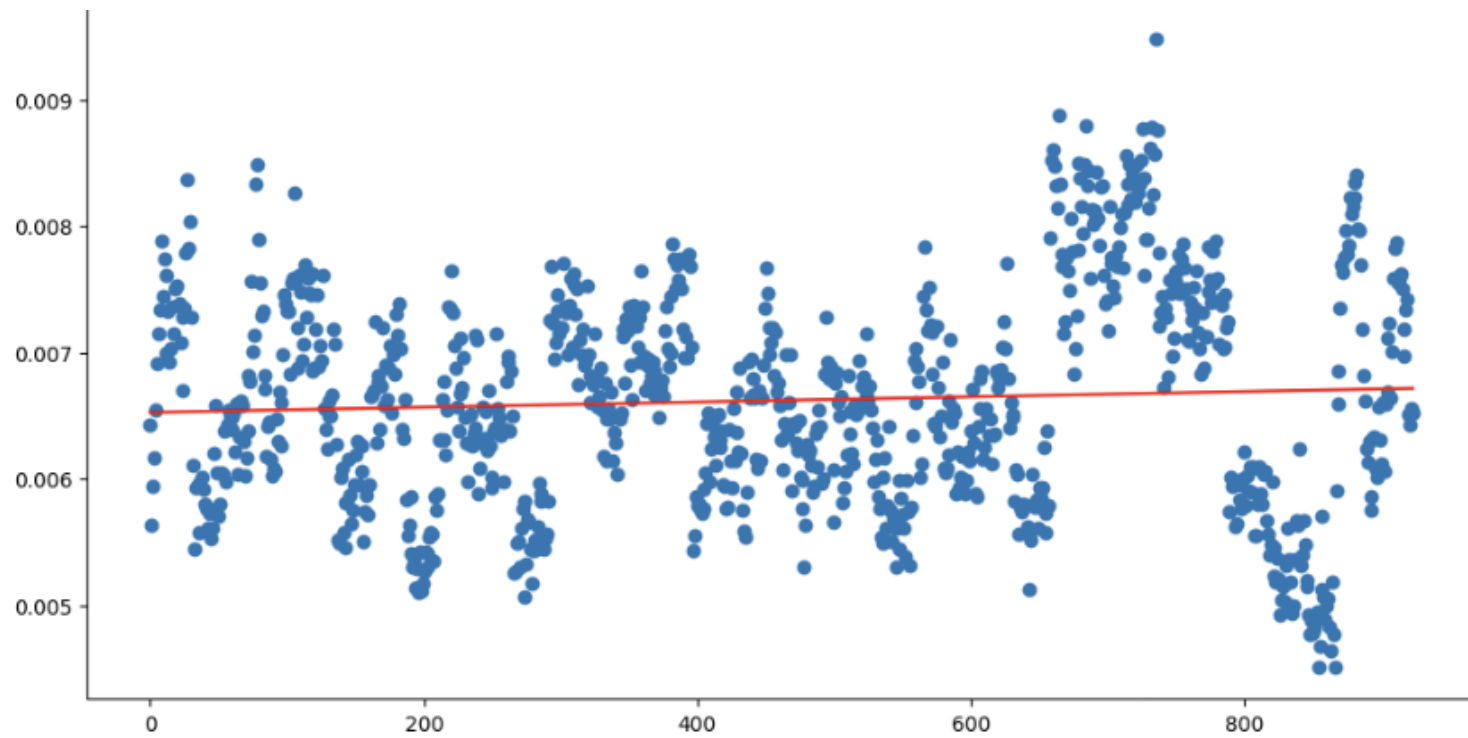
재구성 손실값 계산



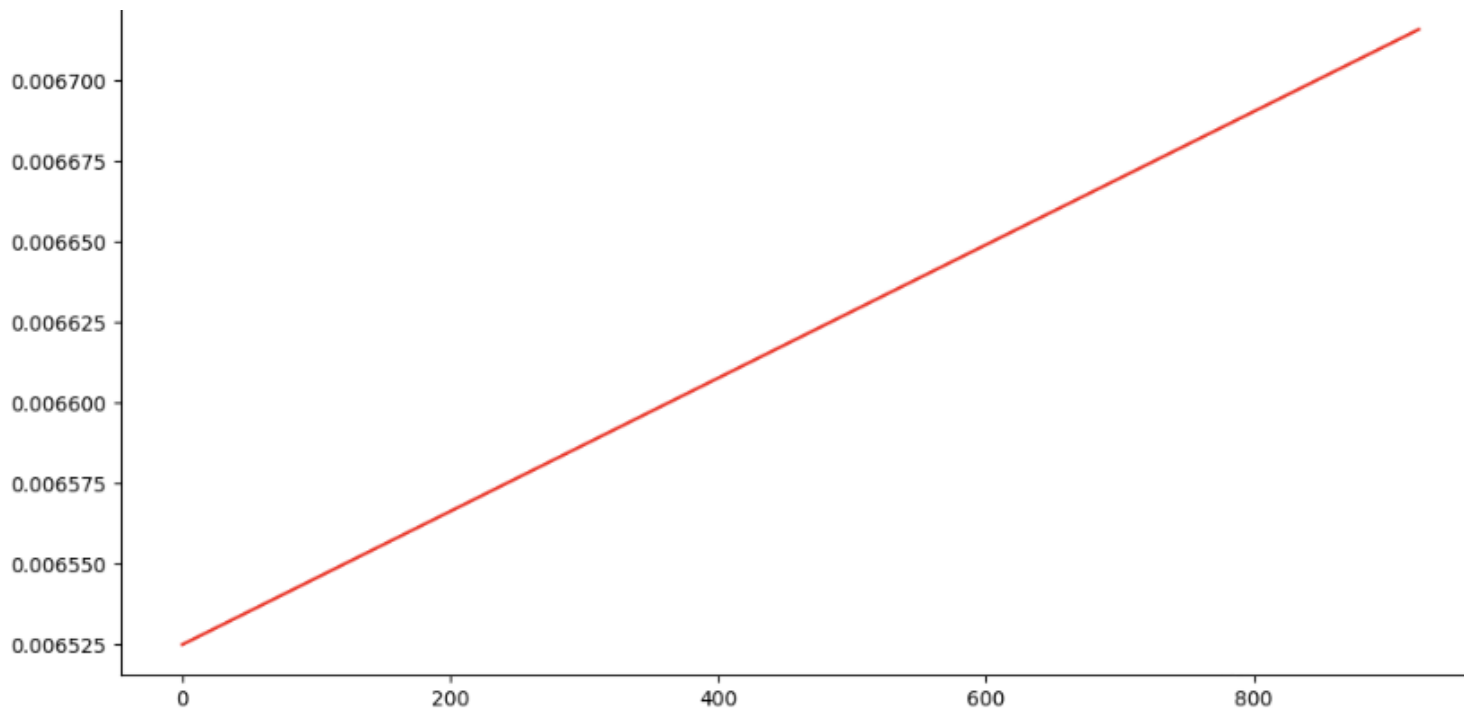
재구성 손실값 계산



정상 데이터의 진동값 변화율



단순화한
정상 데이터



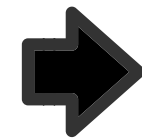
해당 데이터의 추세선

정상 데이터의 진동값 변화율

Imbalance 데이터의 재구성 손실의 재구성 손실값 = 0.0194

- 정상 데이터의 재구성 손실의 재구성 손실값 = 0.0063

0.131

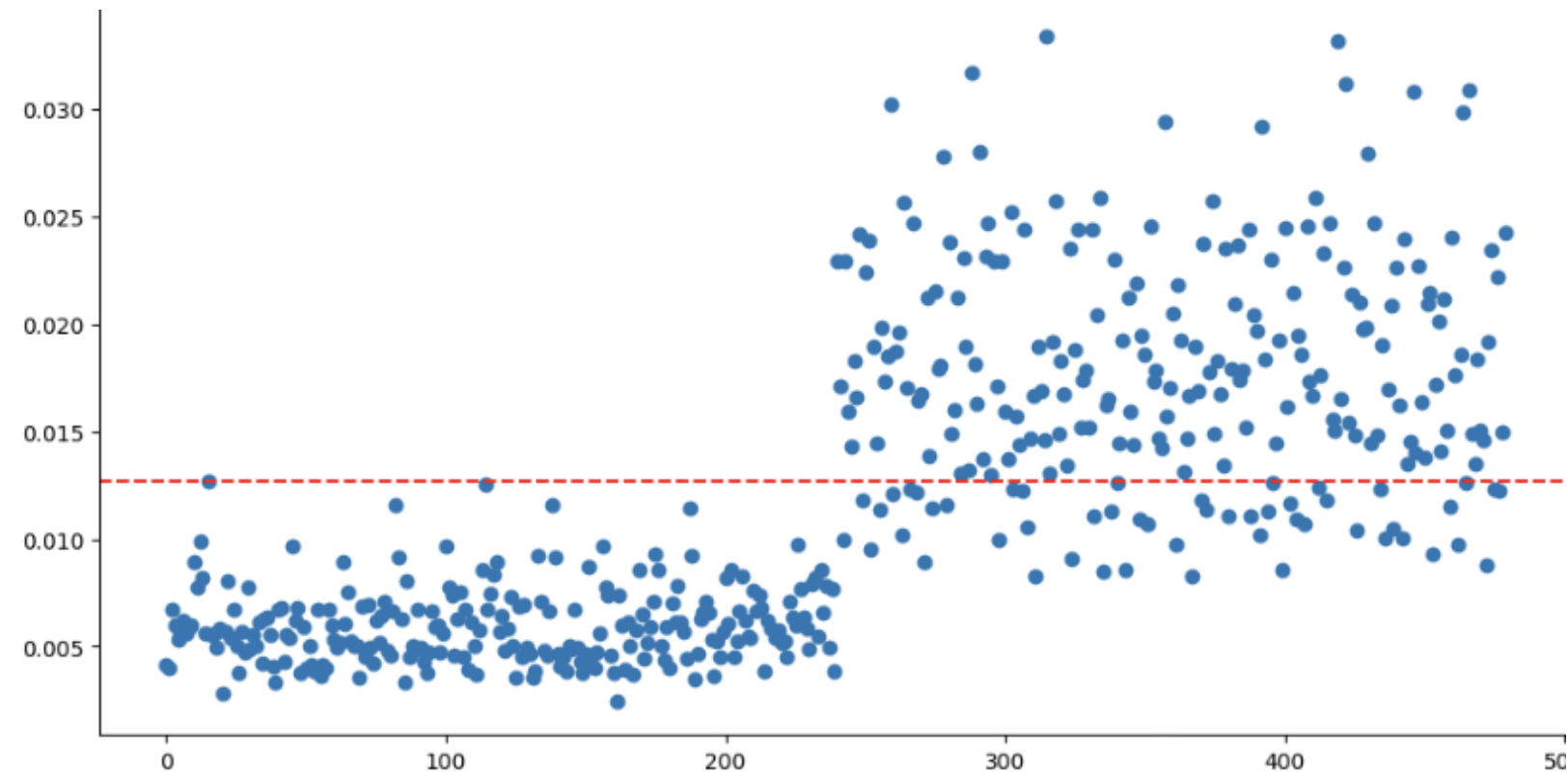


정상 데이터의 재구성 손실의 변화량 = $0.0002 * 65$

공정 부품 예지 고장 서비스



AutoEncoder를 통한 이상상태 분류



정상 진동
데이터

이물질 투입 진동
데이터



전처리 완료

49979개의 데이터 학습 완료

Accuracy: 91.94%

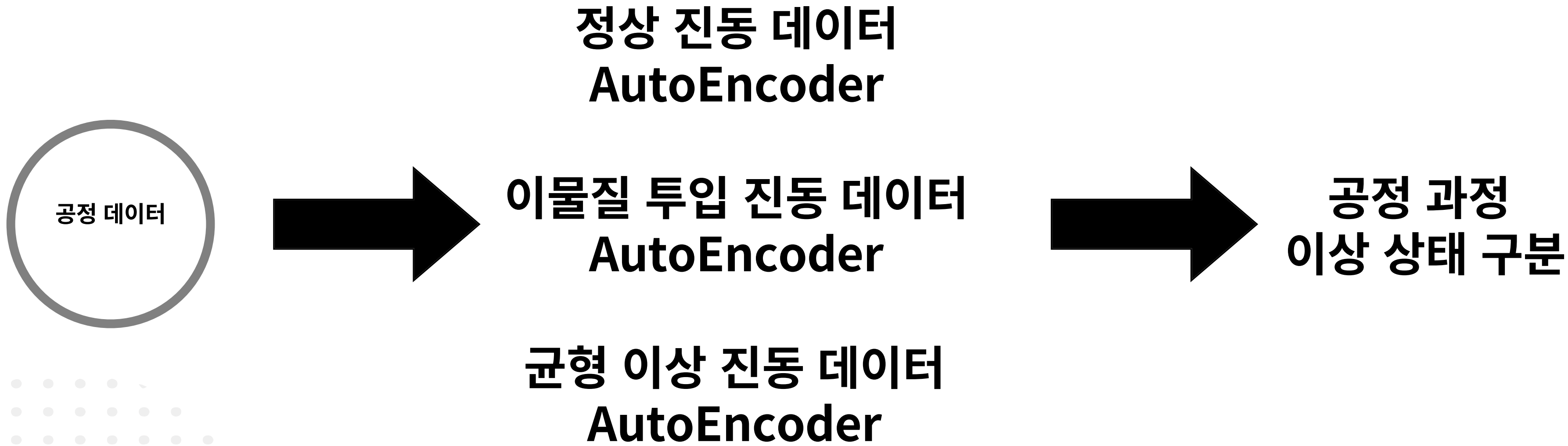
XGBoost Accuracy: 92.50%

Logistic Regression Accuracy: 76.53%

SVM Accuracy: 68.47%

LGBM Accuracy: 91.53%

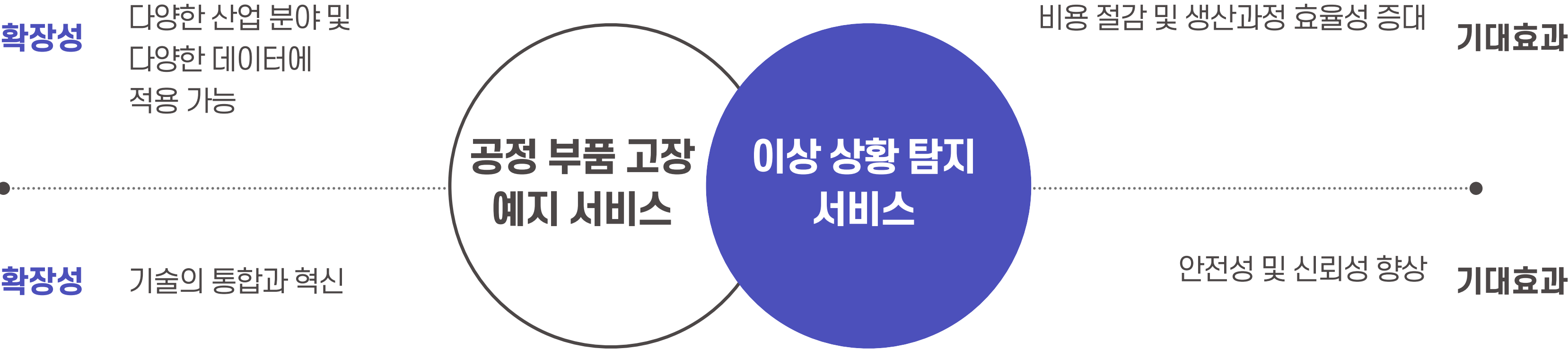
이상 상황 탐지 서비스



03

기대효과

확장성 및 기대효과



2023 경남 제조AI 해커톤 대회

**THANK
YOU**

PROVIBETECH