로봇 팔 모터
이상 징후 감지 시스템 개발

Filab 김창현, 오정윤

Pohang University of Science and Technology

❖ 목차

- S1. Proposal Review

- 문제정의
- 프로젝트 목표
- 진행된 사항

- S2. 알고리즘 개발

- 데이터 전처리
- 이상 탐지
- 알고리즘 적용

- S3. 시스템 개발

- 시스템 기능 및 요구사항
- DB스키마
- 데이터 수집 및 전송 파이프라인
- 서비스

❖ S1. Proposal Review : 문제 정의

문제 정의 로봇 팔 핵심 제어 부품인 **모터의 결함 및 이상 발생 시 전체 시스템이 중단**됨

- ✓ 로봇 팔 : 산업 현장의 반복적이고 정밀한 작업 수행을 위한 다관절 자동화 장비
- ✓ 모터 : 로봇 관절 및 구동 시스템에 동력과 토크를 제어하는 핵심 구동 장치

산업용 로봇팔



로봇 1대의 고장이 라인 전체에 미치는 영향(조선비즈, 2019)

이 기능은 로봇의 집단 지성과 같은 '필드(FIELD) 시스템'으로 이어지고 있다. **그동안 로봇 100대 중 1대가 고장 나면 라인 전체가 중단됐다.** 앞으로 이 시스템이 적용되면 나머지 99대가 공정 속도를 늦춘 뒤 작업을 분담한다. 네트워크로 두뇌를 연결한 로봇이 스스로 공장을 돌리는 것이다. 화낙은 최근 이 시스템으로 720시간 동안 사람 없이 공장을 가동하는 '완전 무인(無人) 공정'에 성공했다.

무리한 로봇 팔 가동으로 인한 고장 사례(동아일보, 2024)

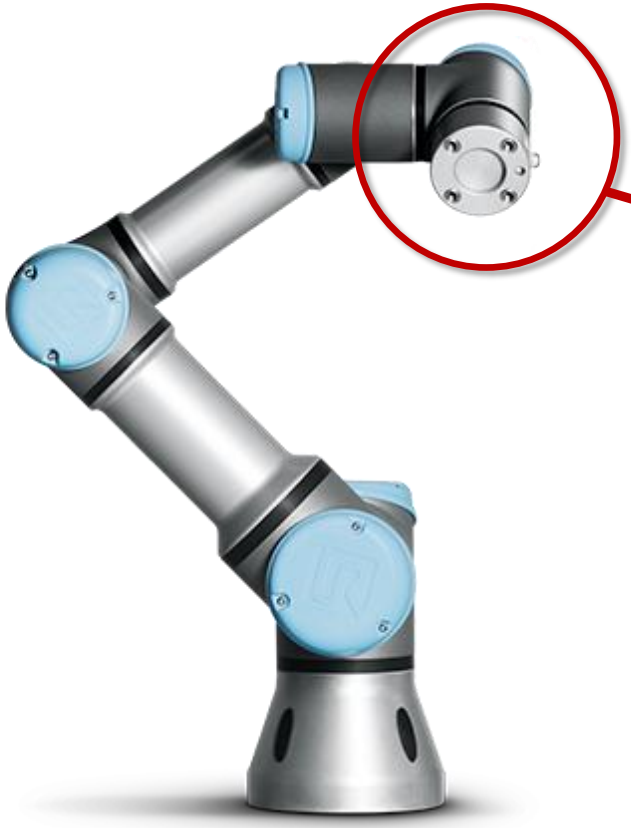
로봇 팔은 원래 고장이 자주 나는 제품이 아니다. 실제 대기업 제조 현장에서는 짧게는 8년, 길게는 10년 이상 사용되곤 한다. 하지만 같은 로봇 팔이라도 중소기업에서 운용될 때 수명은 유독 더 짧다. 그 원인을 성노윤 대표는 '현실과 이상의 차이'로 본다. 그는 "**중소기업에서는 물량과 납기를 맞추기 위해 무리하게 로봇 팔을 가동하기도 한다. 그래서 실제 수명보다 단명하거나 오작동이 잦아지는 것**"이라며 "한 대에 1억 원이 넘는 로봇 팔을 대기업에서는 여분을 두고 여유롭게 운용할 수 있지만, 중소기업이 그러기는 어렵다"고 설명했다.

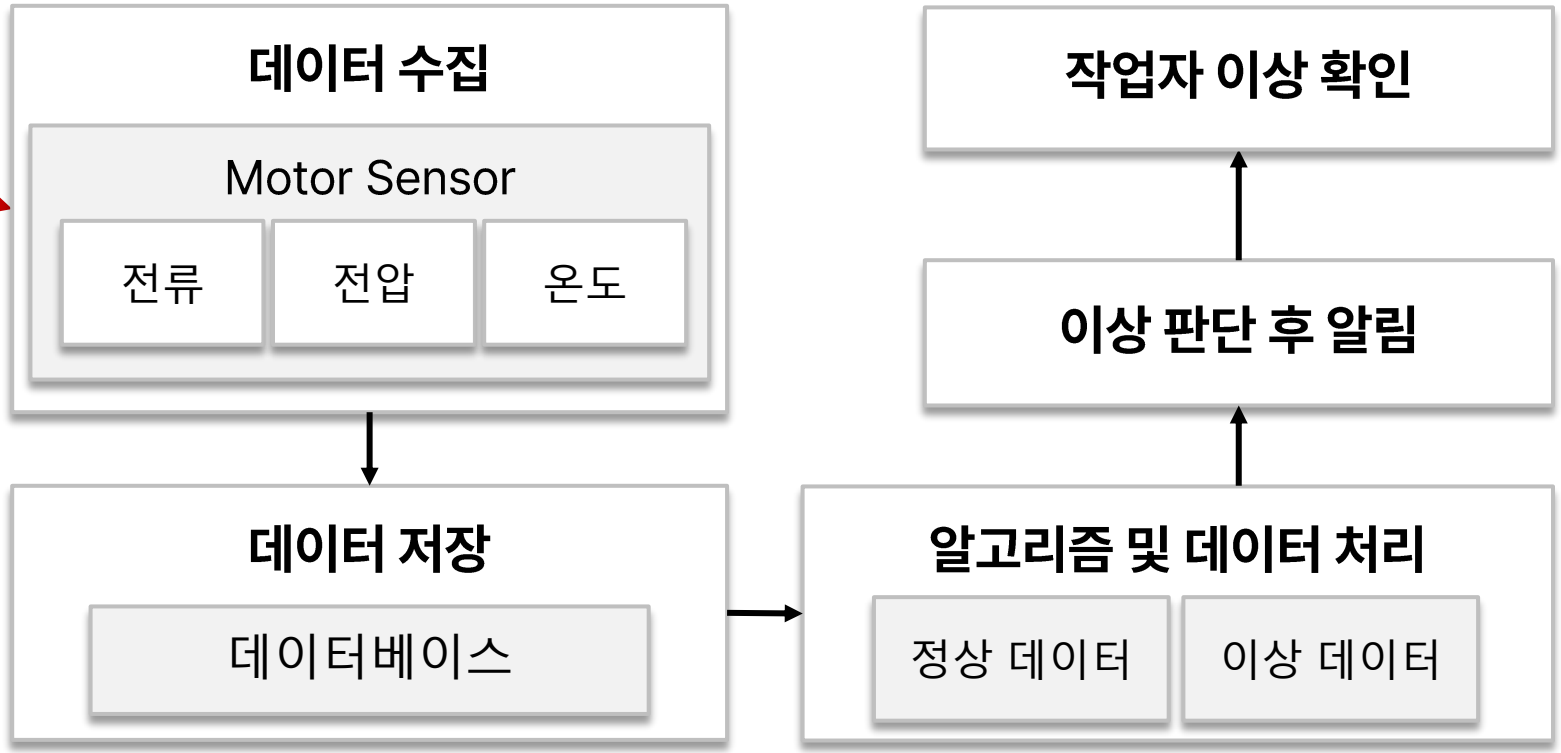
❖ S1. Proposal Review : 과제 목표

과제 목표 로봇 팔 모터 **데이터**를 기반으로 **모니터링 및 이상 감지 시스템**을 구현

✓ 로봇 팔 : 산업 현장의 반복적이고 정밀한 작업 수행을 위한 다관절 자동화 장비

✓ 모터 : 로봇 관절 및 구동 시스템에 동력과 토크를 제어하는 핵심 구동 장치





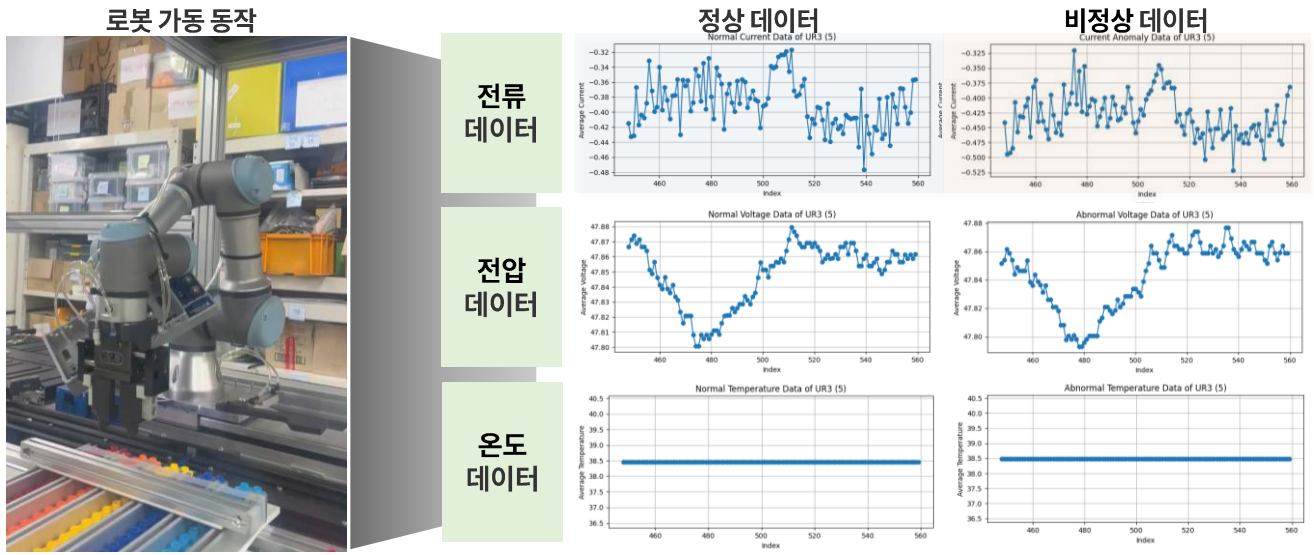
❖ S1. Proposal Review : 진행된 사항

✓ 부품 고장이나 노후로 인한 토크 증가 모사를 위해 무거운 물체 부착 후 이상 데이터 수집

- Cycle time **3.5초**당 **112개**를 **실시간** 수집
- 로봇 끝단에 가중 하중보다 **무거운 하중(3.5kg)을 부과**하여 재현

✓ 수집 데이터 중 사용 가능 데이터 선별 및 정상/이상 라벨링

- 정상과 비정상 간 차이가 식별되지 않는 온도 데이터 제외
- 정상 및 비정상 데이터 라벨링



정상 데이터

timestamp	temp_j1	temp_j2	temp_j3	temp_j4	temp_j5	temp_j6	volt_j1	volt_j2	volt_j3	volt_j4	volt_j5	volt_j6	curr_j1	curr_j2	curr_j3	curr_j4	curr_j5	curr_j6
23:49.2	27.73119	29.56878	29.68943	31.09771	32.68378	30.24345	47.727	47.834	47.91	47.849	47.91	47.818	0.284866	-0.90619	-1.25053	-0.37039	0.128684	-0.02574
23:49.2	27.73119	29.56878	29.68943	31.09771	32.6854	30.24345	47.697	47.834	47.91	47.849	47.94	47.818	0.278137	-0.78731	-1.14988	-0.52033	0.128684	-0.05707
23:49.3	27.73407	29.56878	29.68949	31.09794	32.6854	30.24345	47.697	47.834	47.895	47.864	47.94	47.818	0.132339	-0.55403	-1.06143	-0.50355	0.125327	-0.06266
23:49.3	27.73407	29.56891	29.68949	31.09794	32.68686	30.24811	47.697	47.818	47.895	47.864	47.925	47.834	-0.06056	-0.47777	-0.94705	-0.59083	0.117494	-0.09735
23:49.3	27.73666	29.56891	29.68949	31.09794	32.68686	30.24811	47.727	47.818	47.879	47.864	47.925	47.834	-0.09869	-0.37459	-0.89672	-0.60314	0.113018	-0.03245
23:49.4	27.73666	29.56902	29.68954	31.09814	32.68686	30.25229	47.727	47.818	47.879	47.864	47.925	47.834	-0.25346	-0.47328	-0.78387	-0.57628	0.100709	-0.19359
23:49.4	27.74	29.56902	29.68954	31.09814	32.68918	30.25229	47.727	47.818	47.879	47.864	47.925	47.834	-0.32076	-0.38356	-0.82505	-0.67923	0.06714	-0.07945
23:49.4	27.74	29.56902	29.68958	31.09833	32.68918	30.25229	47.727	47.818	47.925	47.849	47.925	47.834	-0.23103	-0.45085	-0.75185	-0.65014	0.023499	-0.05371
23:49.5	27.743	29.56911	29.68958	31.09833	32.69226	30.25607	47.682	47.834	47.925	47.849	47.925	47.818	-0.21758	-0.41721	-0.73659	-0.63111	-0.03021	-0.09511

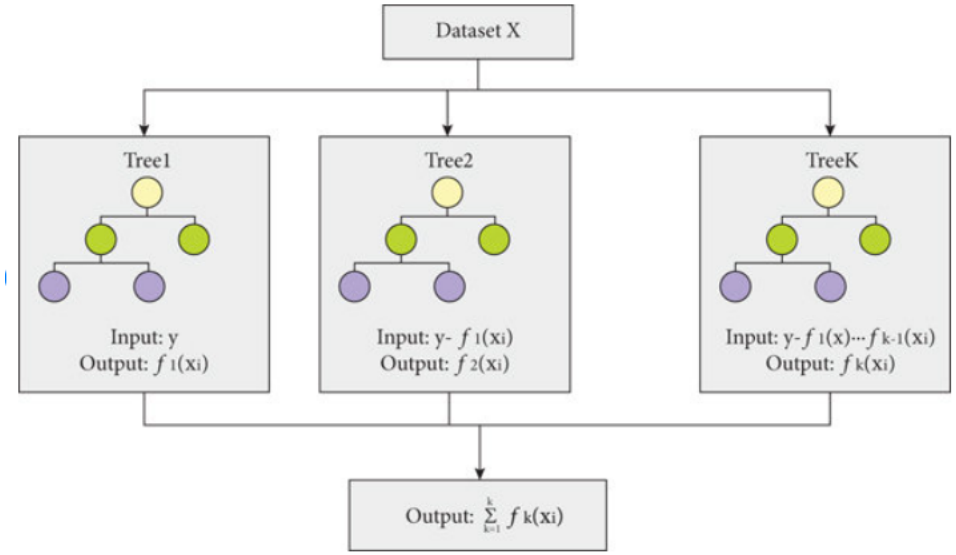
비정상 데이터

timestamp	temp_j1	temp_j2	temp_j3	temp_j4	temp_j5	temp_j6	volt_j1	volt_j2	volt_j3	volt_j4	volt_j5	volt_j6	curr_j1	curr_j2	curr_j3	curr_j4	curr_j5	curr_j6
32:56.5	27.42724	29.93999	30.96972	31.84117	33.85284	31.3442	47.697	47.803	47.895	47.879	47.895	47.834	-0.11215	-0.85909	-1.09193	-0.61545	-0.09623	-0.09511
32:56.5	27.42752	29.93999	30.96972	31.84117	33.85284	31.3442	47.712	47.788	47.895	47.879	47.895	47.834	-0.02467	-0.80974	-0.9135	-0.59866	-0.10966	-0.03805
32:56.6	27.42752	29.93999	30.96972	31.84406	33.85856	31.3442	47.712	47.788	47.895	47.834	47.895	47.834	-0.11215	-0.60786	-1.06905	-0.61657	-0.13316	-0.16337
32:56.6	27.42777	29.93999	30.96974	31.84406	33.85856	31.34578	47.682	47.818	47.91	47.834	47.895	47.818	-0.1929	-0.60114	-0.92417	-0.56397	-0.14211	-0.10966
32:56.6	27.42777	29.93999	30.96974	31.84406	33.85856	31.34578	47.682	47.818	47.91	47.834	47.91	47.818	-0.323	-0.58095	-1.01415	-0.65349	-0.15554	-0.05259
32:56.7	27.42777	29.93999	30.96977	31.84665	33.8637	31.3482	47.697	47.818	47.91	47.834	47.91	47.834	-0.26917	-0.65721	-0.9257	-0.59866	-0.17344	-0.06826
32:56.7	27.42799	29.93999	30.96977	31.84665	33.8637	31.3482	47.697	47.818	47.91	47.834	47.91	47.834	-0.2086	-0.61459	-1.00805	-0.59419	-0.16002	-0.10295

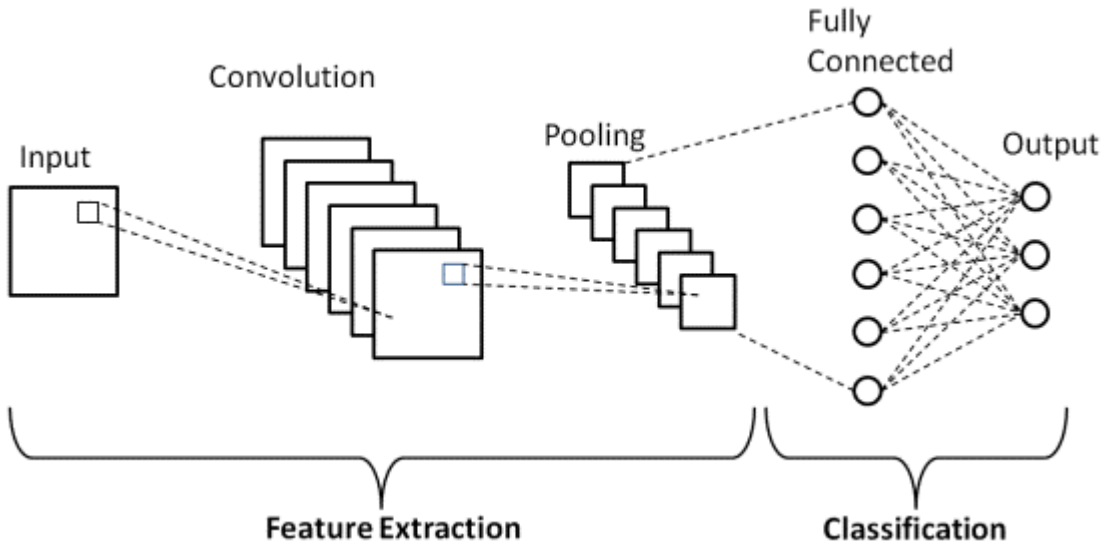
❖ S2. 알고리즘 개발

수집한 데이터를 바탕으로 **이상 움직임**에 대해 **파악**할 수 있는 **AI 모델** 제작

- ✓ 한 동작 (cycle)에 대해 각각 ML/DL 방법론을 적용한 모델을 만들어 모델의 적합성 비교 진행
- ✓ **ML 모델**: Boosting 기반의 **XGBoost, LightBGM**
- ✓ **DL 모델**: **CNN**



Boosting based XGBoost

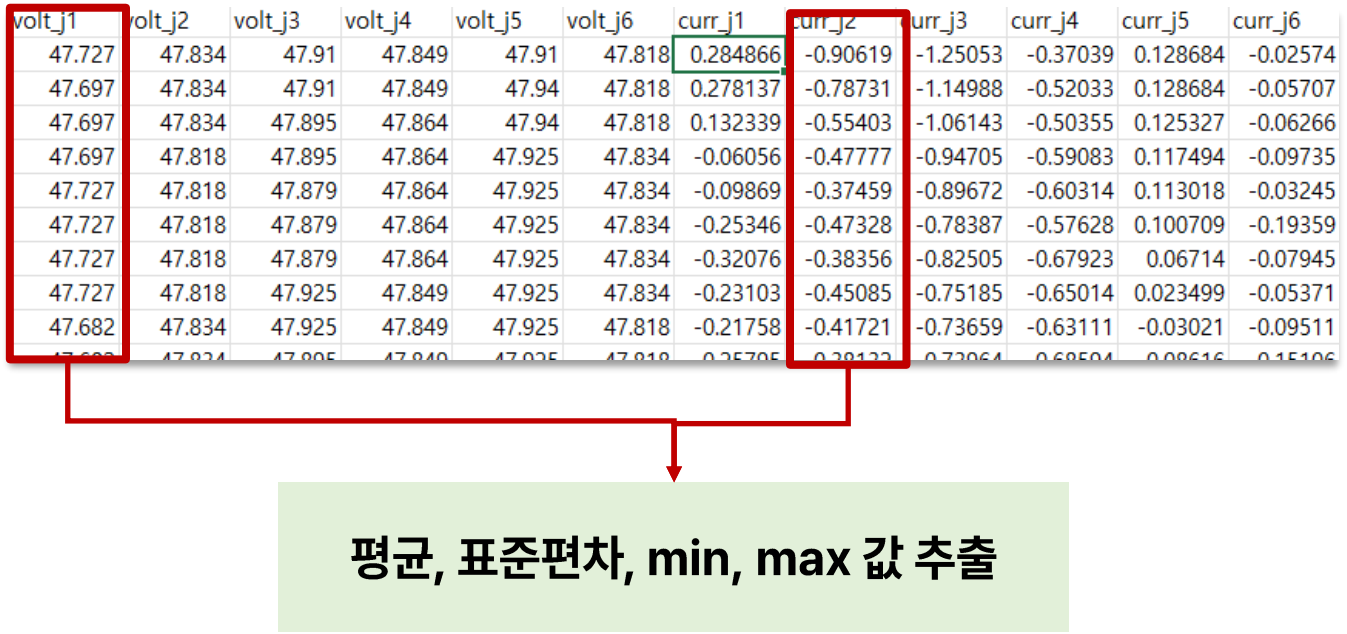


CNN

❖ S2. 알고리즘 개발 : 데이터 전처리

하나의 cycle에서 저장되는 **2차원 [112*12]** 사이즈의 데이터 → **[1*48]의 벡터**로 데이터 요약함

✓ 각각의 column에서 [평균, 표준편차, min, max]값 추출하여 하나의 벡터로 이어 총 12개의 column에서 각각 4개의 통계 값이 반영된 **[1*48]의 벡터**로 전처리



volt_j1	volt_j2	volt_j3	volt_j4	volt_j5	volt_j6	curr_j1	curr_j2	curr_j3	curr_j4	curr_j5	curr_j6
47.727	47.834	47.91	47.849	47.91	47.818	0.284866	-0.90619	-1.25053	-0.37039	0.128684	-0.02574
47.697	47.834	47.91	47.849	47.94	47.818	0.278137	-0.78731	-1.14988	-0.52033	0.128684	-0.05707
47.697	47.834	47.895	47.864	47.94	47.818	0.132339	-0.55403	-1.06143	-0.50355	0.125327	-0.06266
47.697	47.818	47.895	47.864	47.925	47.834	-0.06056	-0.47777	-0.94705	-0.59083	0.117494	-0.09735
47.727	47.818	47.879	47.864	47.925	47.834	-0.09869	-0.37459	-0.89672	-0.60314	0.113018	-0.03245
47.727	47.818	47.879	47.864	47.925	47.834	-0.25346	-0.47328	-0.78387	-0.57628	0.100709	-0.19359
47.727	47.818	47.879	47.864	47.925	47.834	-0.32076	-0.38356	-0.82505	-0.67923	0.06714	-0.07945
47.727	47.818	47.925	47.849	47.925	47.834	-0.23103	-0.45085	-0.75185	-0.65014	0.023499	-0.05371
47.682	47.834	47.925	47.849	47.925	47.818	-0.21758	-0.41721	-0.73659	-0.63111	-0.03021	-0.09511

평균, 표준편차, min, max 값 추출

❖ S2. 알고리즘 개발 : 이상 탐지

- **Input**: 앞서 전처리 진행한 벡터 데이터 (정상 데이터: 88개, 비정상 데이터: 87개)
- **Train data**: 80% / test data: 20%로 나눠 모델 학습

➔ 두 모델 모두 정상/비정상 데이터를 **완벽하게 탐지**하며 **높은 성능**을 보여줌

모델 적용 parameters

	XGBoost	LightGBM
트리 개수	100	100
learning_rate	0.1	0.1
max_depth	5	-1(제한 X)
boosting_type	-	gbdt
eval_metric	logloss	-

모델별 탐지 정확도

	Accuracy
XGBoost	100%
LightGBM	100%

정상, 비정상 데이터 완벽 탐지

❖ S2. 알고리즘 개발 : 이상 탐지

통계량 기반 학습 → 신호 데이터의 **구조적 패턴 훼손 및 시간적 순서가 사라짐**

전처리 및 피쳐 엔지니어링 수행이 필요없는 DL 방식 사용 : **CNN**

- ✓ 한 cycle 내에 존재하는 [112*12]사이즈의 데이터를 **2D data**로 인식
- ✓ 각 cycle 단위를 **one-shot 단위**로 구분해 전체 움직임에 대해 정상/비정상으로 나눔

volt_j1	volt_j2	volt_j3	volt_j4	volt_j5	volt_j6	curr_j1	curr_j2	curr_j3	curr_j4	curr_j5	curr_j6
47.727	47.834	47.91	47.849	47.91	47.818	0.284866	-0.90619	-1.25053	-0.37039	0.128684	-0.02574
47.697	47.834	47.91	47.849	47.94	47.818	0.278137	-0.78731	-1.14988	-0.52033	0.128684	-0.05707
47.697	47.834	47.895	47.864	47.94	47.818	0.132339	-0.55403	-1.06143	-0.50355	0.125327	-0.06266
47.697	47.818	47.895	47.864	47.925	47.834	-0.06056	-0.47777	-0.94705	-0.59083	0.117494	-0.09735
47.727	47.818	47.879	47.864	47.925	47.834	-0.09869	-0.37459	-0.89672	-0.60314	0.113018	-0.03245
47.727	47.818	47.879	47.864	47.925	47.834	-0.25346	-0.47328	-0.78387	-0.57628	0.100709	-0.19359
47.727	47.818	47.879	47.864	47.925	47.834	-0.32076	-0.38356	-0.82505	-0.67923	0.06714	-0.07945
47.727	47.818	47.925	47.849	47.925	47.834	-0.23103	-0.45085	-0.75185	-0.65014	0.023499	-0.05371
47.682	47.834	47.925	47.849	47.925	47.818	-0.21758	-0.41721	-0.73659	-0.63111	-0.03021	-0.09511
47.682	47.834	47.895	47.849	47.925	47.818	-0.25795	-0.38132	-0.73964	-0.68594	-0.08616	-0.15106
47.712	47.834	47.895	47.864	47.925	47.818	-0.26917	-0.3387	-0.7793	-0.72175	-0.09176	-0.14211
47.712	47.834	47.895	47.864	47.925	47.818	-0.3387	-0.48225	-0.72439	-0.64118	-0.14211	-0.18687
47.712	47.834	47.925	47.849	47.925	47.818	-0.25571	-0.44412	-0.74575	-0.73518	-0.11526	-0.11861

Test Loss	0.1849
Test Accuracy	0.9714

❖ S2. 알고리즘 개발 : 이상 탐지

이슈 : ML/DL 기반의 모델은 전체 한 cycle이 진행되고 나서야 진단할 수 있다는 이슈 존재

- ➢ **Input**: 정상적 움직임 data가 저장되는 과정에 이상 움직임을 부여해 **정상/비정상 데이터가 섞인** 데이터 활용 (정상 : 비정상 = 80% : 20%, 총 9958개 데이터)
- ➢ **Train data**: 80% / test data: 20%로 나눠 모델 학습.
- ✓ LSTM 레이어는 2개가 포함됨

➔ **연속적 내용을 반영**하는 모델 구조로 더 높은 실시간 성능 확보

CNN 탐지 정확도

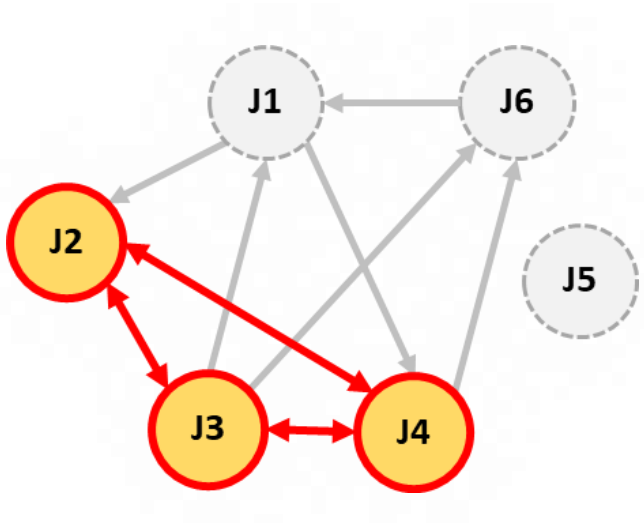
Test Loss	0.1849
Test Accuracy	0.9574

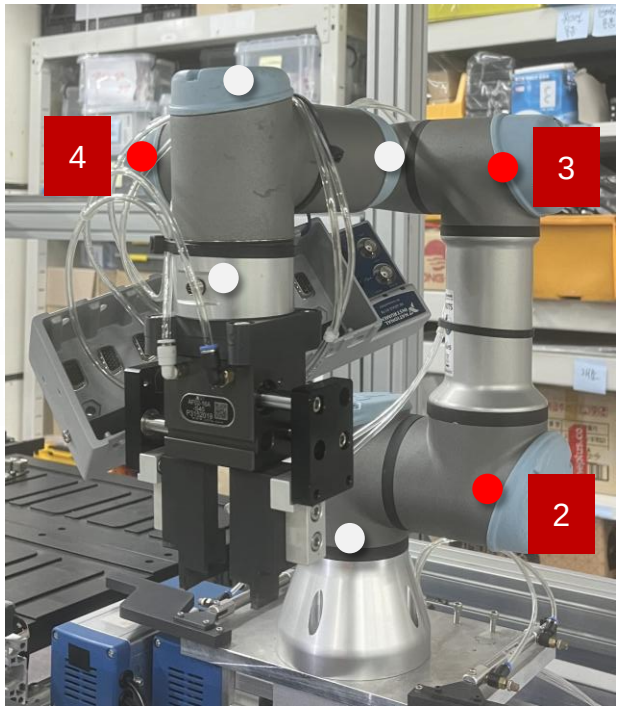
LSTM 탐지 정확도

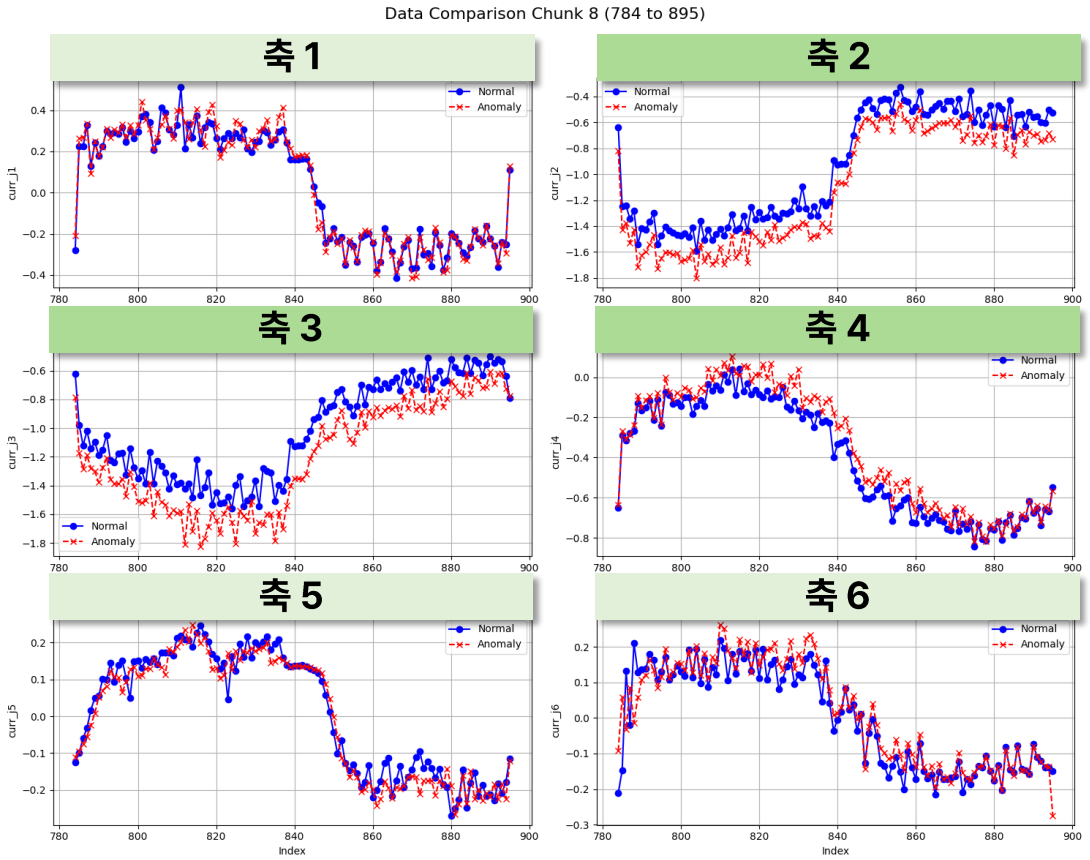
Test Loss	0.0772
Test Accuracy	0.9784

❖ S2. 알고리즘 개발 : 알고리즘 적용

- AI 모델의 경량화를 위해 분석 중요도가 높은 데이터만을 선별하여 활용하고자 함
- 각 축별 데이터에 대한 **그레인저 인과성 검정** 결과, **2, 3, 4번** 축에 대한 인과성이 높게 분석됨







❖ S2. 알고리즘 개발 : 알고리즘 적용

- **Input**: 전체 데이터 중 영향력이 큰 **2, 3, 4번 축에서 측정되는 전압, 전류** 데이터만 활용함
  - 사용 column의 개수: 12개 → 6개
  - 나머지 세팅에 대해선 기존 모델과 동일하게 진행

➔ 입력 변수의 수를 **절반으로 줄여** LSTM 모델의 복잡도를 낮추고 연산 효율성을 개선함

➔ 정확도 측면에서 여전히 높은 수준을 유지함

➔ 계산 비용을 줄여 **경량화 및 빠른 예측이 가능**해졌다는 점에서 실시간 시스템에 더욱 적합

Light한 버전의 탐지 정확도

Test Accuracy	0.9272

≈

LSTM 탐지 정확도

Test Accuracy	0.9784

❖ S2. 알고리즘 개발 : 알고리즘 적용

- 학습한 LSTM 모델을 로봇 프로그램에 적용하여, 실시간 이상탐지 수행
- **Time window**: 10, Stride: 1로 설정
- **모델 추론 시간**: 약 0.04초



문제 발견

비정상 움직임 발생시,

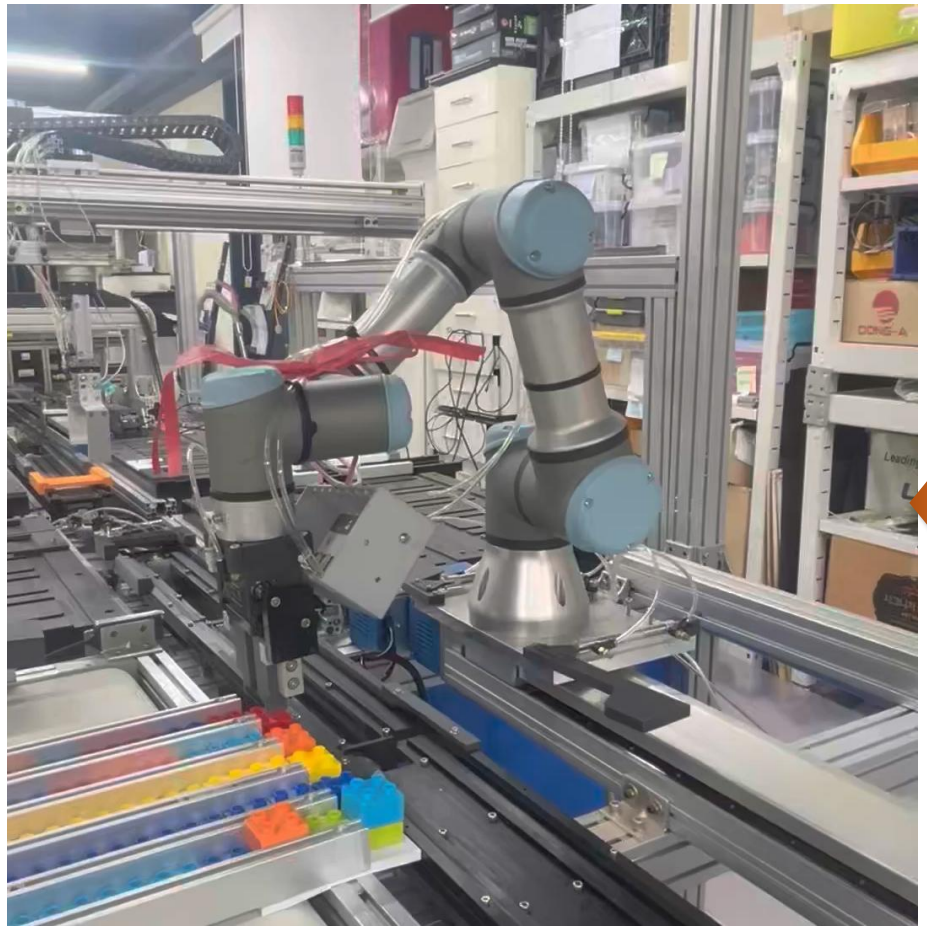
로봇이 정지하는데 딜레이가 크게 존재

데이터 수집 주기가 0.03초이나, 학습한 모델로부터

결과를 도출하는데 약 **0.04초의 시간이 소요**

❖ S2. 알고리즘 개발 : 알고리즘 적용

➢ 데이터 수집 주기 변경 후, 비정상 움직임 포착 시 **로봇의 즉각적 대응 성공**



해결 방안

데이터 수집 주기를 **0.06초로 변경** 후,

새로 수집한 데이터로 **모델 재학습**하여 문제 해결

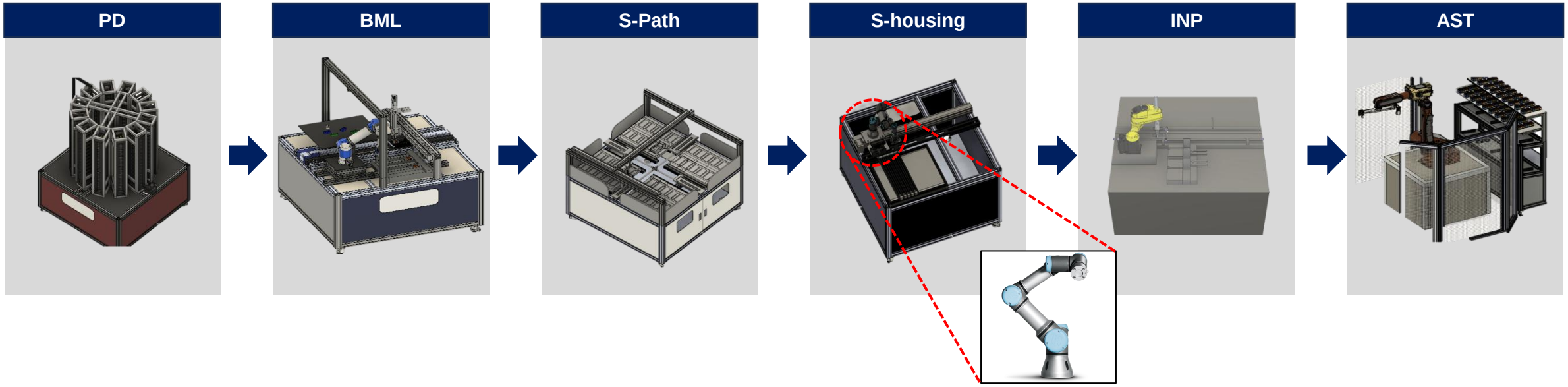
❖ S3. 시스템 개발 : 시스템 기능 및 요구사항

워크셀 제조공정

- ➢ 제조 라인은 공정 목적에 따라 여러 개의 워크셀로 구성되어 있음
- ➢ 각 워크셀마다 설치된 로봇 및 설비 종류가 다양함

➔ 설비 운영 중 발생하는 이상을 놓치게 될 경우 **타 워크셀도 정지되어 생산성이 저하**됨

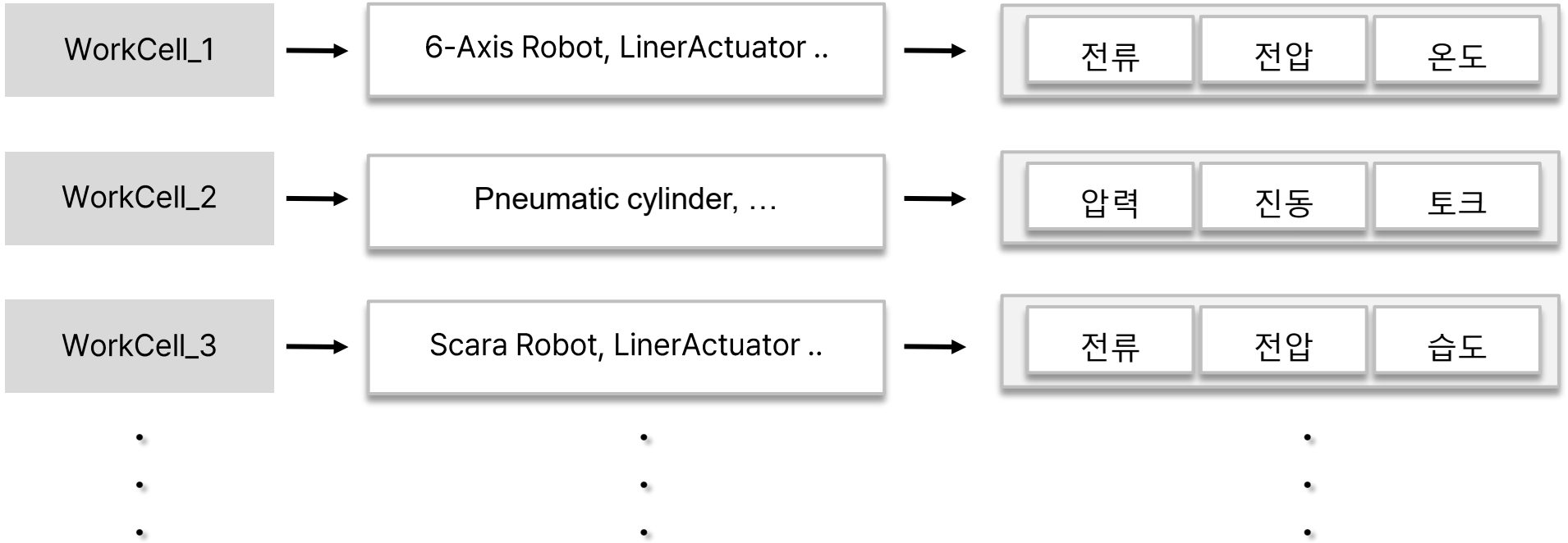
➔ 로봇 및 설비 종류가 달라 설비 이상 탐지에 **필요한 데이터 종류 및 특성이 동일하지 않음**



❖ S3. 시스템 개발 : 시스템 기능 및 요구사항

로봇 팔 모터 이상징후 감지 시스템 개발 고려사항

- ➢ 실시간성(Real-time) : 로봇 신호를 Edge에서 즉시 추론 및 이상 대응
- ➢ 확장성(Scalable) : 워크셀 혹은 로봇 늘어나도 동일 구조로 확장
- ➢ 스키마 유연성(Schema Evolution) : 센서 채널, 모델 출력 등이 늘어나는 상황 대응 가능

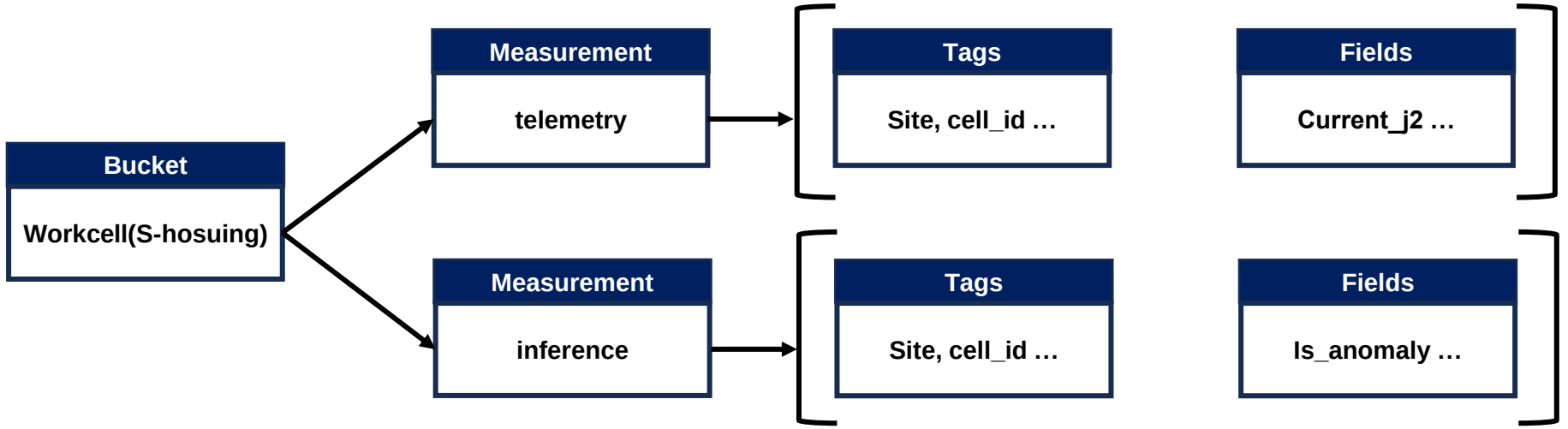


S3. 시스템 개발 : DB 스키마

TSDB(Time Series DataBase) 중 하나인 InfluxDB를 사용

- Bucket : DB와 유사(location with retention policy)
- Measurement : RDB의 table과 유사
- Tags : 필터 & 그룹핑에 쓰는 식별자, 인덱스 컬럼
- Fields : 인덱싱 되지 않는 컬럼

RDB	InfluxDB
Database	Database
Table	Measurement
Indexed column	Tag key
Unindexted column	Filed key
Row	Point



❖ S3. 시스템 개발 : DB 스키마

예시

- ➢ Bucket : S-hosuing
- ➢ Measurement : telemetry(raw-data)
- ➢ Tags : site(라인 이름), cell_id(워크셀 id), asset_type(장비 종류), asset_id(장비 번호)
- ➔ FactoryLab 라인의 cellA 워크셀 안의 UR 로봇 1번 장비
- ➢ Fields : current_j2, current_j3, ... voltage_j4

_time	_value	_field	_measurement	asset_id	asset_type	cell_id	site
2025-12-07 06:07:06 GMT+9	-1.13	current_j2	telemetry	UR3_01	UR	cellA	FactoryLab
2025-12-07 06:07:07 GMT+9	-1.36	current_j2	telemetry	UR3_01	UR	cellA	FactoryLab
2025-12-07 06:07:08 GMT+9	-1.43	current_j2	telemetry	UR3_01	UR	cellA	FactoryLab
2025-12-07 06:07:09 GMT+9	-1.34	current_j2	telemetry	UR3_01	UR	cellA	FactoryLab
2025-12-07 06:07:10 GMT+9	-1.27	current_j2	telemetry	UR3_01	UR	cellA	FactoryLab
2025-12-07 06:07:11 GMT+9	-0.59	current_j2	telemetry	UR3_01	UR	cellA	FactoryLab

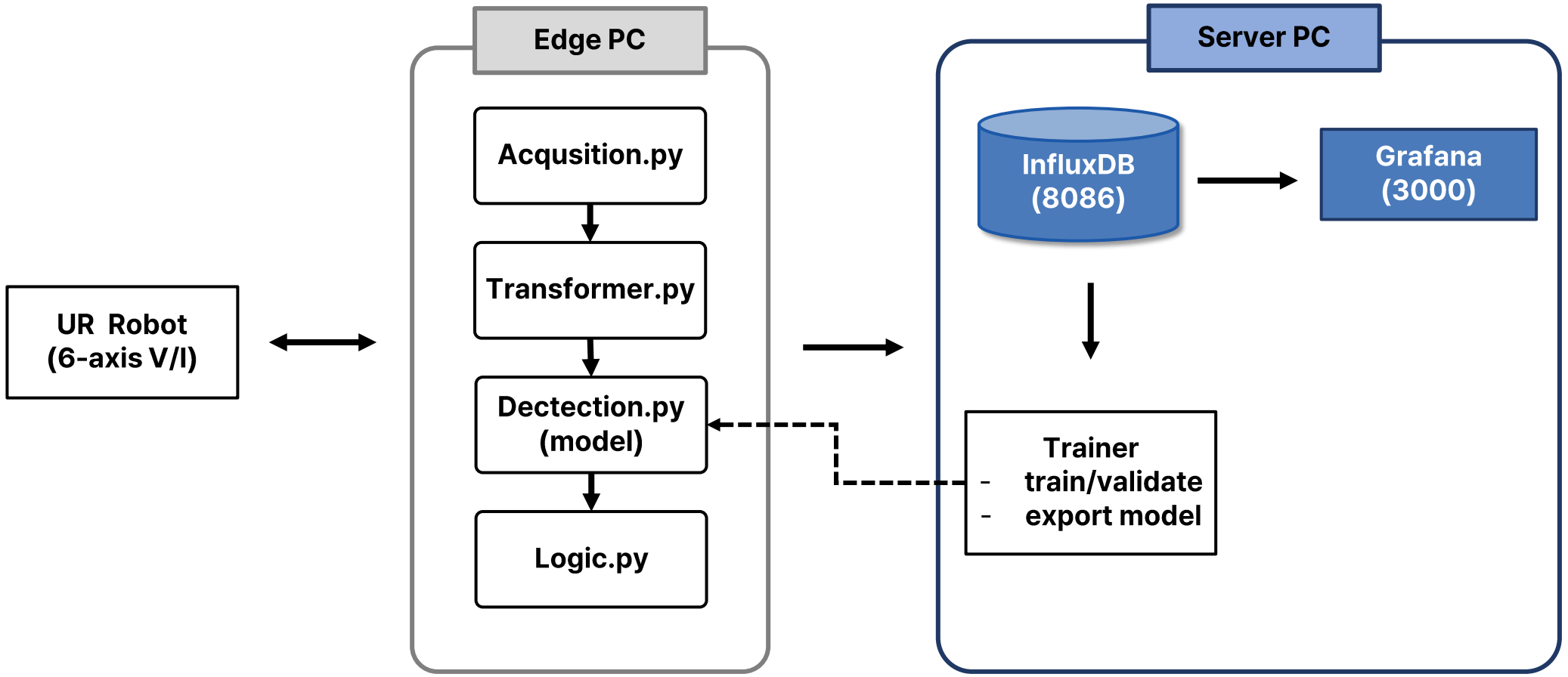
❖ S3. 시스템 개발 : DB 스키마

예시

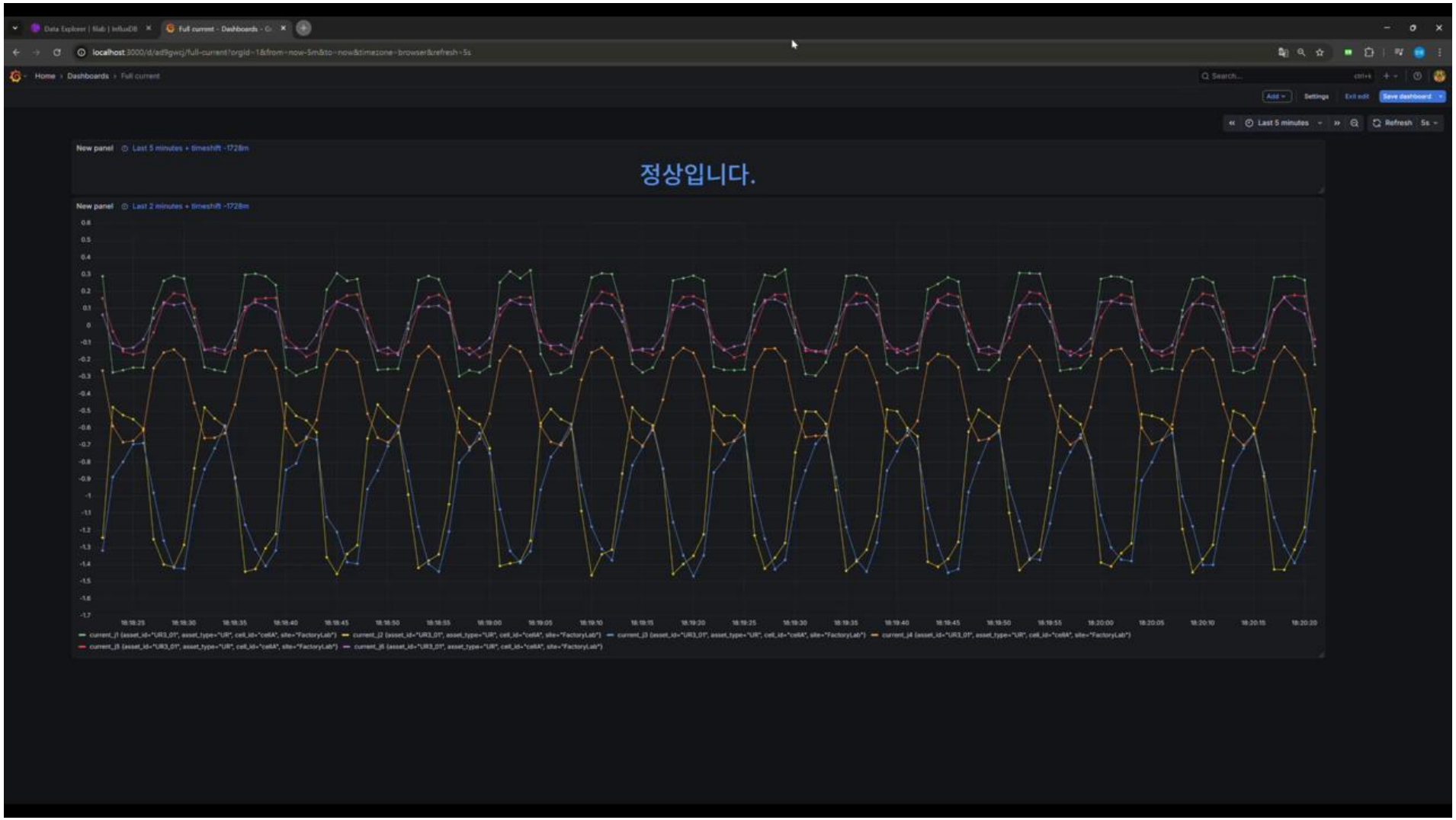
- ➢ Bucket : S-hosuing
- ➢ Measurement : inference(추론 결과 값)
- ➢ Tags : site(라인 이름), cell_id(워크셀 id), asset_type(장비 종류), asset_id(장비 번호)
- ➔ FactoryLab 라인의 cellA 워크셀 안의 UR 로봇 1번 장비
- ➢ Fields : is_anomaly

_time	_value	_field	_measurement	asset_id	asset_type	cell_id	site	model	model_ver
2025-12-07 06:07:06 G...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0
2025-12-07 06:07:06 G...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0
2025-12-07 06:07:07 GM...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0
2025-12-07 06:07:07 GM...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0
2025-12-07 06:07:07 GM...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0
2025-12-07 06:07:07 GM...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0
2025-12-07 06:07:07 GM...	0	is_anomaly	inference	UR3_01	UR	cellA	FactoryLab	ur_anom_v1	1.0.0

❖ S3. 시스템 개발 : 데이터 수집 및 전송 파이프라인



❖ S3. 시스템 개발 : 서비스


Full current - Dashboards
localhost:3000/d/ad9gwsj/full-current?orgId=1&from=now-5m&to=now&timezone=browser&refresh=5s
Home › Dashboards › Full current
Add
Settings
Exit edit
Save dashboard
Last 5 minutes
Refresh 5s
New panel
Last 5 minutes + timeshift -1728m
정상입니다.
New panel
Last 2 minutes + timeshift -1728m

감사합니다.