프로젝트 수행 결과보고서

스마트 모니터링 시스템

- 정진우

목차

I. 프로젝트 개요

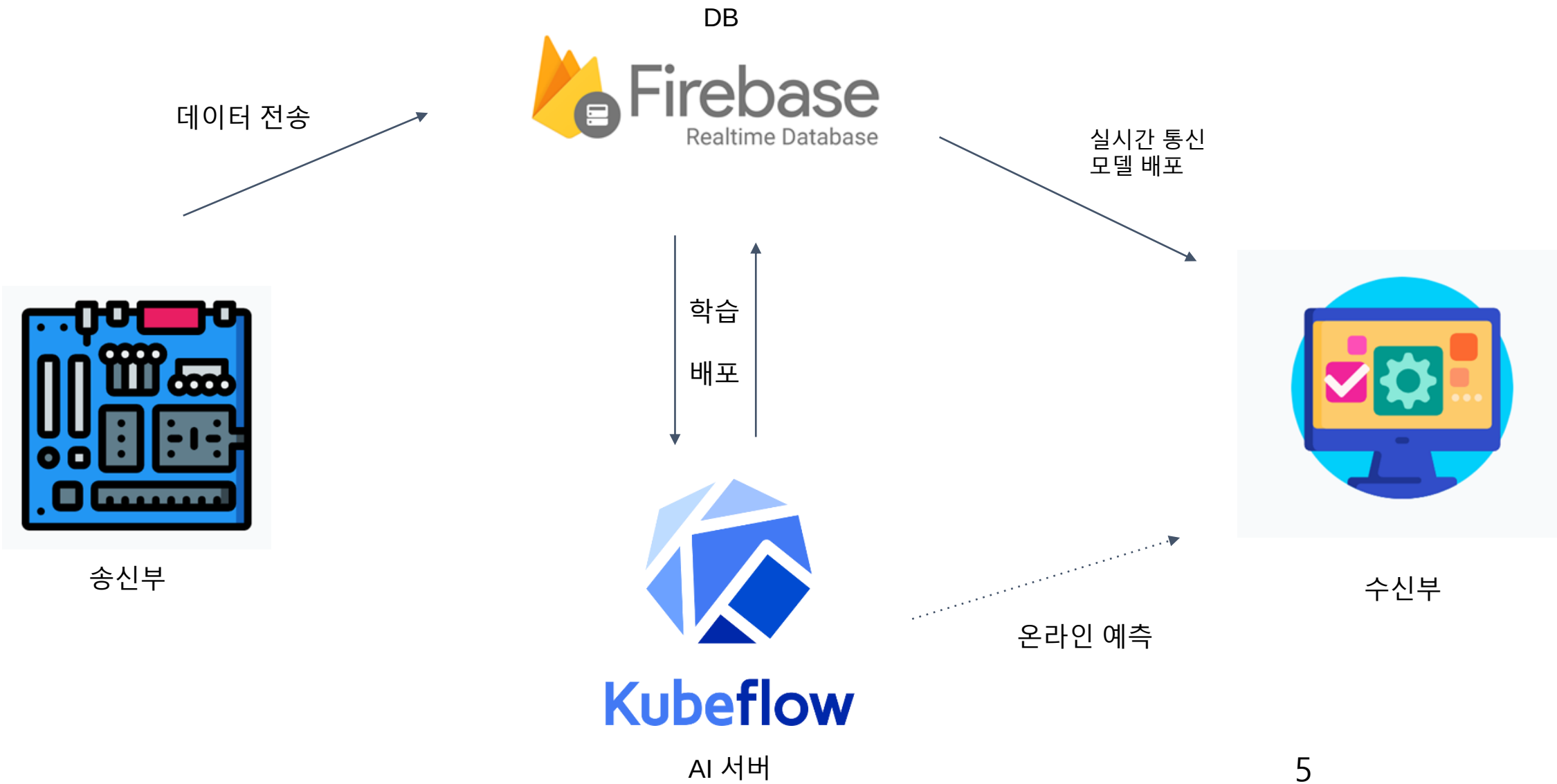
I. 프로젝트 개요

스마트모니터링 시스템

- 송신부 : 각 설비에 IoT(사물인터넷)를 도입하여 전력, 온도, CPA, 수율 등 을 실시간으로 수집한다.
- **DB : 수집된 데이터를 체계적으로 저장한다.**
- **AI 서버 : AI모델을 학습하고 배포한다.**
- **수신부 : 수집된 데이터를 바탕으로 설비의 현재상태, 남은수명, 고장위치의 시각화 등의 정보를 표현**

송신부를 제외한 DB와 AI서버, 수신부를 구현

I. 프로젝트 개요


DB
Firebase
Realtime Database
데이터 전송
실시간 통신
모델 배포
학습
배포
송신부
수신부
Kubeflow
AI 서버
온라인 예측

I. 프로젝트 개요

기본 요구사항

- 설비 현재상태 시각화
- 설비 수명 예측
- 설비 고장 시각화

추가 개발 목표

사용 편의성(End2End 시스템)

- 기본 사용자를 비숙련자로 정의, 필요시 세부정보 제공
- 충분한 시각화로 사용자 조작 필요성 최소화

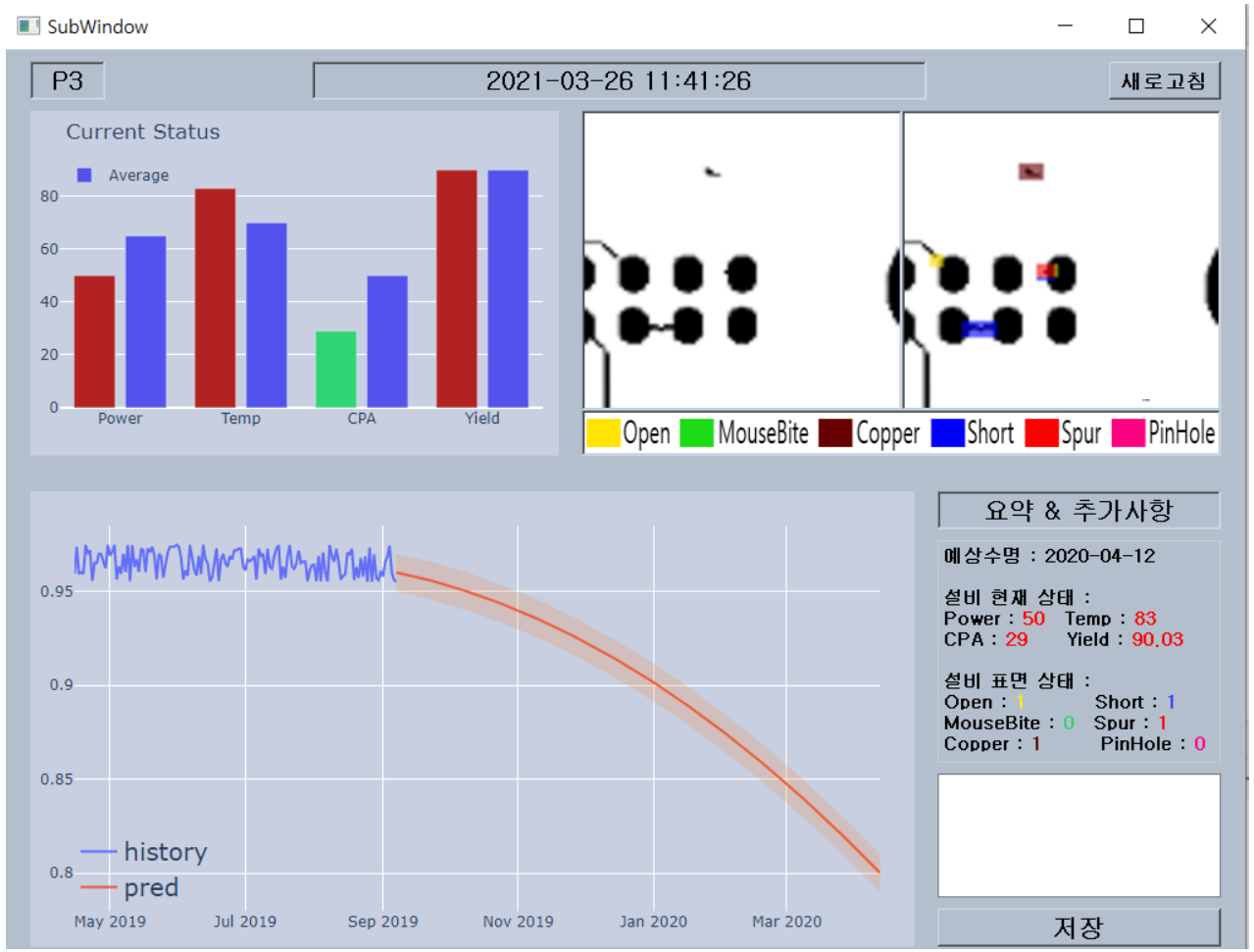
협업, 유지보수, 확장성

- 기능과 목적에 따라 코드를 모듈화
- MVC 디자인패턴을 활용하여 개발 구역을 분할
- AI 학습서버를 활용하여 체계적인 학습, 시각화와 배포

Ⅱ. 프로젝트 기능

Ⅱ. 프로젝트 기능

기본화면



보조화면

Ⅱ. 프로젝트 기능

기본화면

- 현재 설비값을 DB에서 읽어와 각 측정값이 정상/비정상인지를 수치와 색깔로 표현
- 이미지 클릭 시 보조화면으로 상세정보를 표시
- 자동 갱신

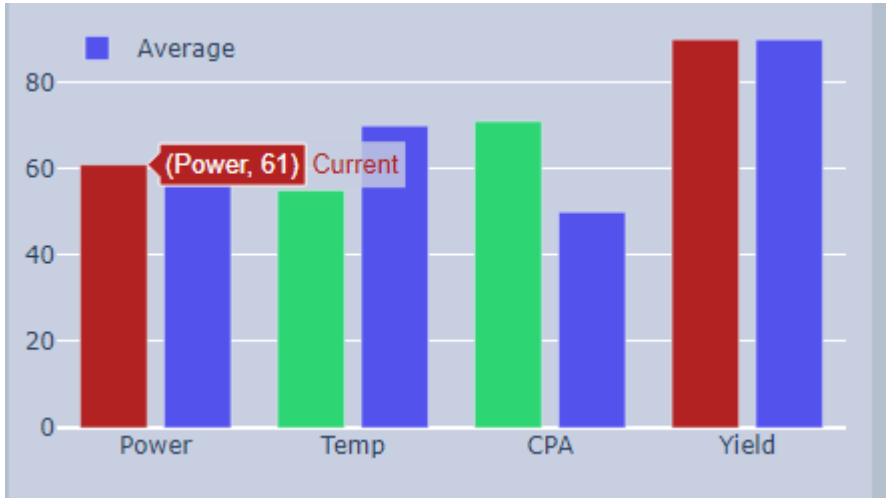
Ⅱ. 프로젝트 기능

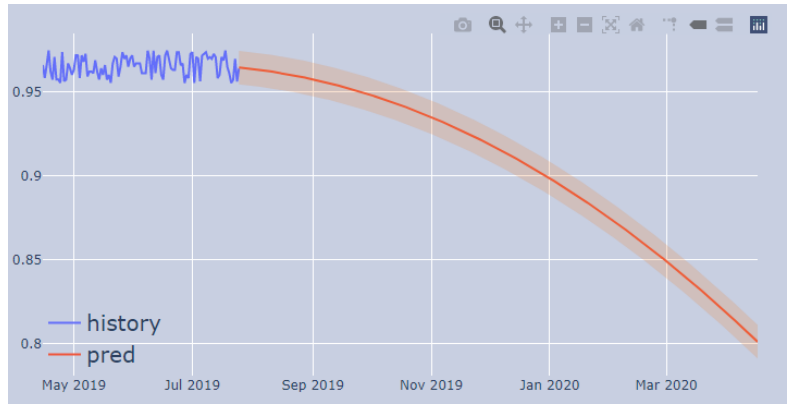
보조화면

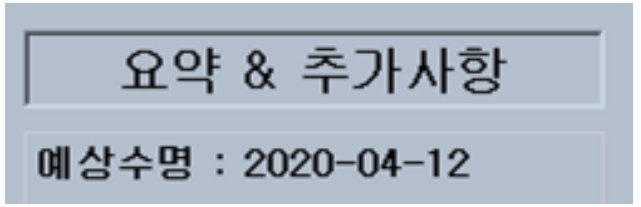
클릭된 설비의 상세정보를 수동갱신

1. 현재 설비의 정보를 막대그래프로 시각화
2. 설비이미지에서 불량위치와 종류를 시각화
3. 설비의 미래수명 예측을 그래프로 표현
4. 전체 정보 요약&추가메모
5. 엑셀로 저장

Ⅱ. 프로젝트 기능



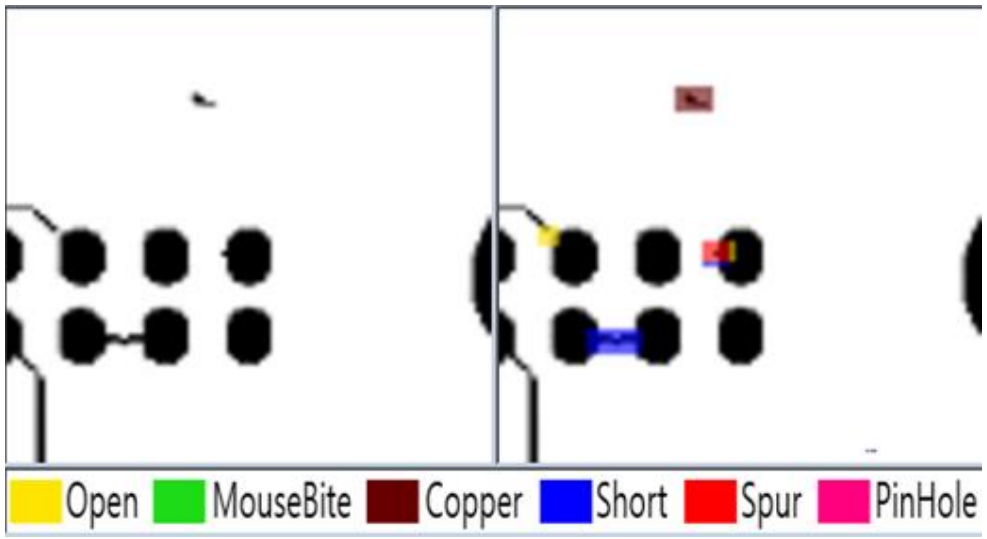




현재 설비의 데이터와 기준치를 비교하여 막대그래프로 표현

머신러닝을 활용하여
현재까지의 기록을 분석하여 남은 수명을 계산

Ⅱ. 프로젝트 기능



설비 표면 상태 :
Open : 1 Short : 1
MouseBite : 0 Spur : 1
Copper : 1 PinHole : 0

- 설비 이미지(왼쪽)에서 고장위치를 분석하여 시각화
- 각 색별로 다른 고장을 표현

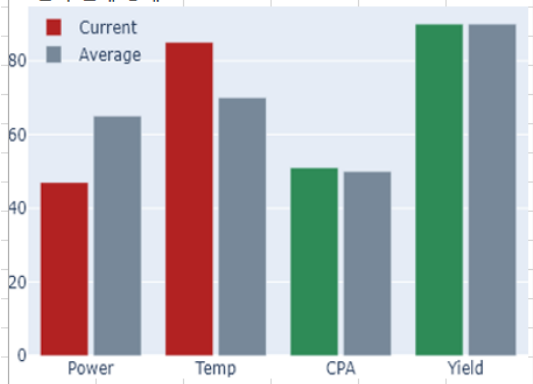
1. 화면의 모든정보를 요약하여 글로 표현
2. 필요시 사용자가 추가로 메모를 하여 저장할 수 있게함
3. 모든 정보는 엑셀로 저장 가능

Ⅱ. 프로젝트 기능

P3

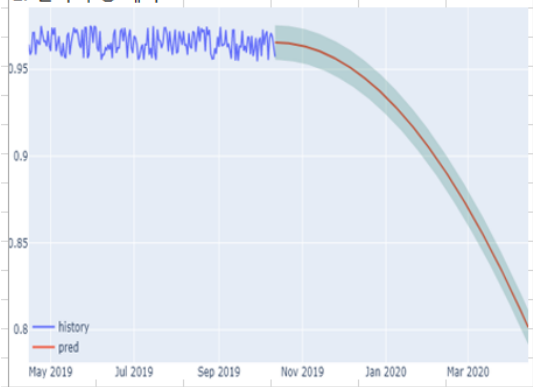
2021-03-25 09:59:34

1. 설비 현재 상태



	전력	온도	CPA	수율
현재	47	85	51	90
평균	65	70	50	90

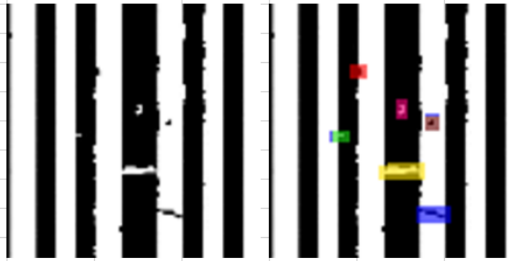
2. 설비 수명 예측



설치일	2019-04-14	가동일수	182일

3. 설비이미지 분석

현재 이미지



AI 분석 결과

Open	Short	MouseBit	Spur	Copper	PinHole
1	1	1	1	1	1

4. 추가사항

점심

Summary | History

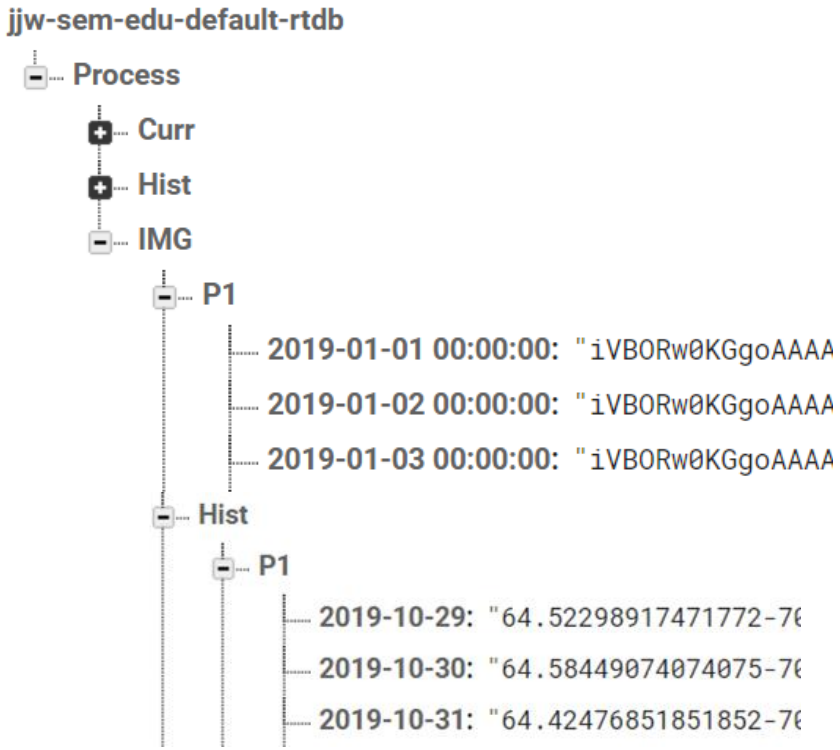
P3

Date	Power	Temp	CPA	Yield	Elapsed
2019-04-14	64.53727126	69.88522605	49.50174919	0.971976605	0
2019-04-15	64.503125	70.1119213	49.40497685	0.964345324	1
2019-04-16	64.54016204	69.95289352	49.37951389	0.958485995	2
2019-04-17	64.47268519	69.93761574	49.44571759	0.960396246	3
2019-04-18	64.49293981	69.96342593	49.53796296	0.971033828	4
2019-04-19	64.49398148	70.02083333	49.47569444	0.971061502	5
2019-04-20	64.58530093	69.85891204	49.56516204	0.958457444	6
2019-04-21	64.53171296	69.9994213	49.35289352	0.967197319	7
2019-04-22	64.5087963	70.05833333	49.46516204	0.965289827	8
2019-04-23	64.42916667	70.05358796	49.47986111	0.964295034	9
2019-04-24	64.54027778	70.05983796	49.61261574	0.974944335	10
2019-04-25	64.48136574	69.97951389	49.40694444	0.970072967	11
2019-04-26	64.53055556	70.04456019	49.55104167	0.966247478	12
2019-04-27	64.47650463	69.90960648	49.62731481	0.966176028	13
2019-04-28	64.48078704	70.08229167	49.49351852	0.963351555	14
2019-04-29	64.42465278	70.11851852	49.51747685	0.972044858	15
2019-04-30	64.40810185	69.99768519	49.56828704	0.963306394	16
2019-05-01	64.44421296	70.25381944	49.52905093	0.96052462	17
2019-05-02	64.47025463	70.10810185	49.44699074	0.973981573	18
2019-05-03	64.45393519	70.01284722	49.35891204	0.970078232	19
2019-05-04	64.55046296	69.90069444	49.44803241	0.973907382	20
2019-05-05	64.52743056	70.05219907	49.41180556	0.960454768	21
2019-05-06	64.45972222	70.1619213	49.54351852	0.964345219	22
2019-05-07	64.57986111	69.87175926	49.56643519	0.969083325	23
2019-05-08	64.48472222	69.99305556	49.52881944	0.960417581	24
2019-05-09	64.52337963	69.95914352	49.54456019	0.957519083	25
2019-05-10	64.58078704	69.92384259	49.57858796	0.965242569	26
2019-05-11	64.50011574	69.94988426	49.40231481	0.957506166	27

Summary | History

Ⅲ. 프로젝트 상세

Ⅲ. 프로젝트 상세



Firebase

- 어플리케이션 개발을 위한 구글의 High Level Platform

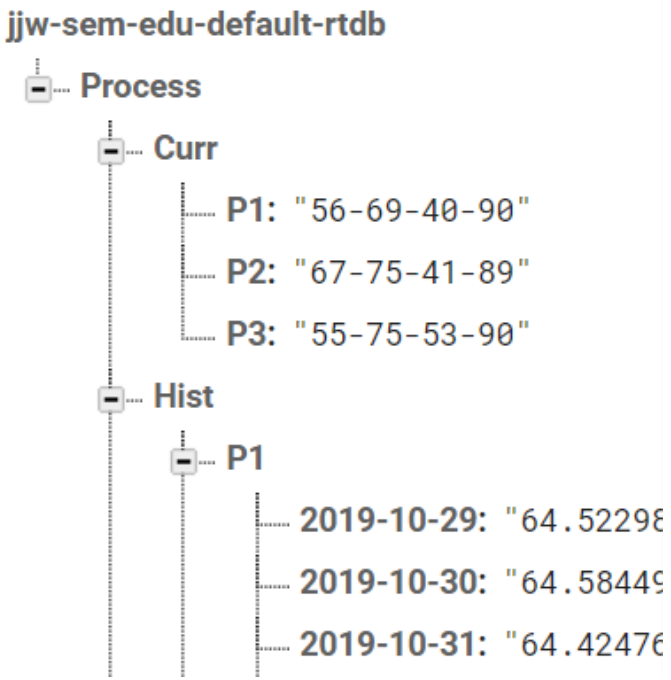
Firebase Database

- 서버를 관리할 필요가 없다.
- 동적인 서버확장.
- Google의 다른 클라우드 서비스와 결합 용이(Kuberflow를 통한 AI 모델학습, 배포)

Ⅲ. 프로젝트 상세

Date	Power	Temp	CPA	Yield
2019-10-04	62	58	50	0.959925
2019-10-04	66	67	44	0.964101
2019-10-04	59	71	51	0.964156
2019-10-04	67	61	58	0.961865
2019-10-04	67	72	53	0.966533
...	...	...	...	...
2019-10-22	65	73	54	0.816496
2019-10-22	65	65	52	0.813079
2019-10-22	64	75	44	0.817144
2019-10-22	64	65	48	0.812877
2019-10-22	66	62	53	0.812081

DataFrame



FireBase RealTime DataBase

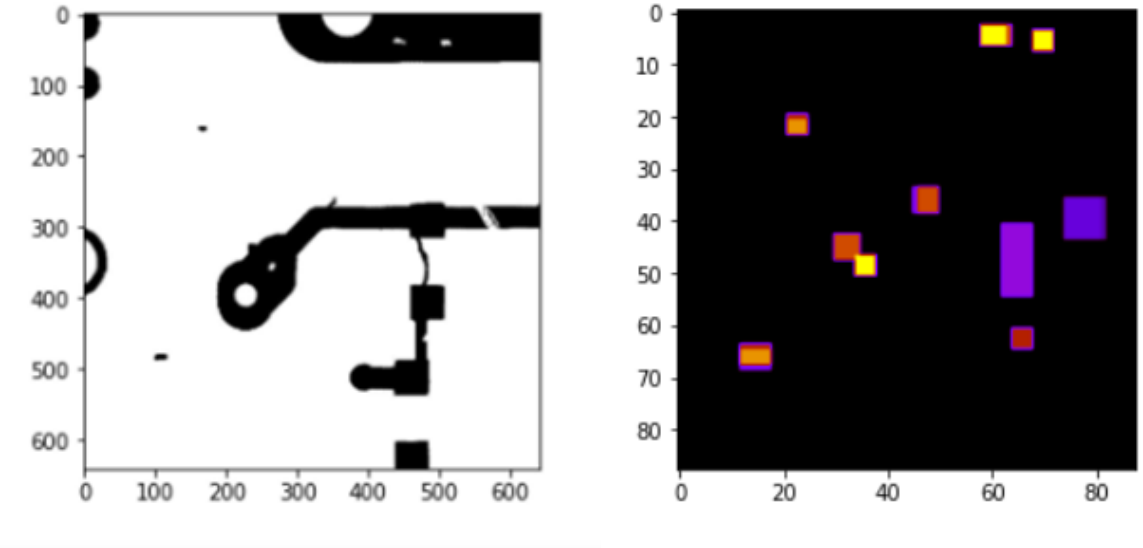
기본 데이터
- 전력(Power), 온도(Temp), CPA, 수율(Yield)로 구성
- 수율은 시간이 시간이 지날수록 떨어지게 설정

데이터베이스
- 최근데이터 : 한 폴더에 한값만 넣어서 빠르게 엑세스
- 데이터기록 : 설비별로 엑세스 가능

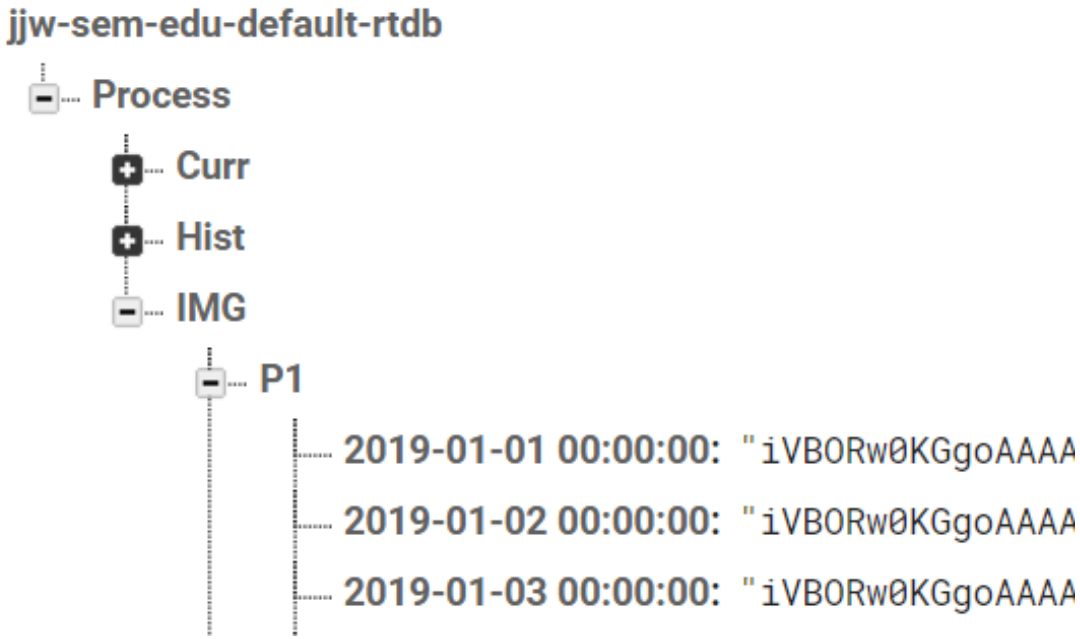
전처리
- 가동일로부터의 경과일수 설정
- 전력,온도,CPA의 누적합, 누적분산, 누적첨도와 왜도를 설정

훈련
- 누적데이터로부터 최종날짜까지의 남은 일수 예측

Ⅲ. 프로젝트 상세



Kaggle DeepPCB Dataset

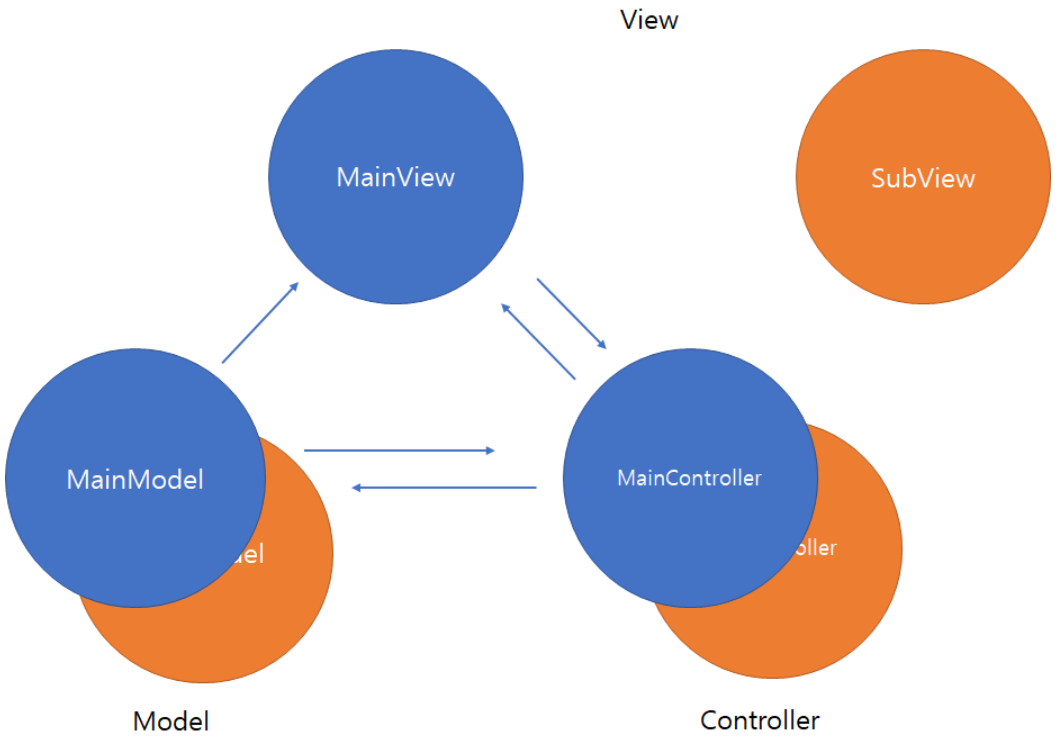


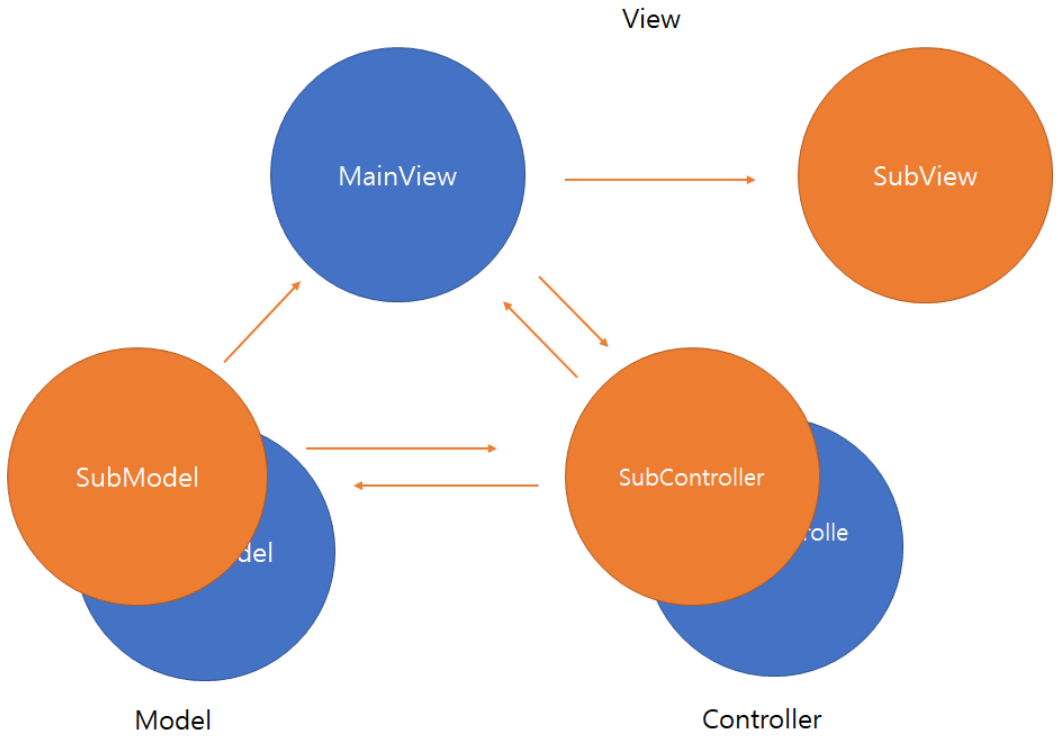
FireBase RealTime DataBase

크기조정->바이트형식->문자열 으로 이미지 저장

Ⅲ. 프로젝트 상세

MVC



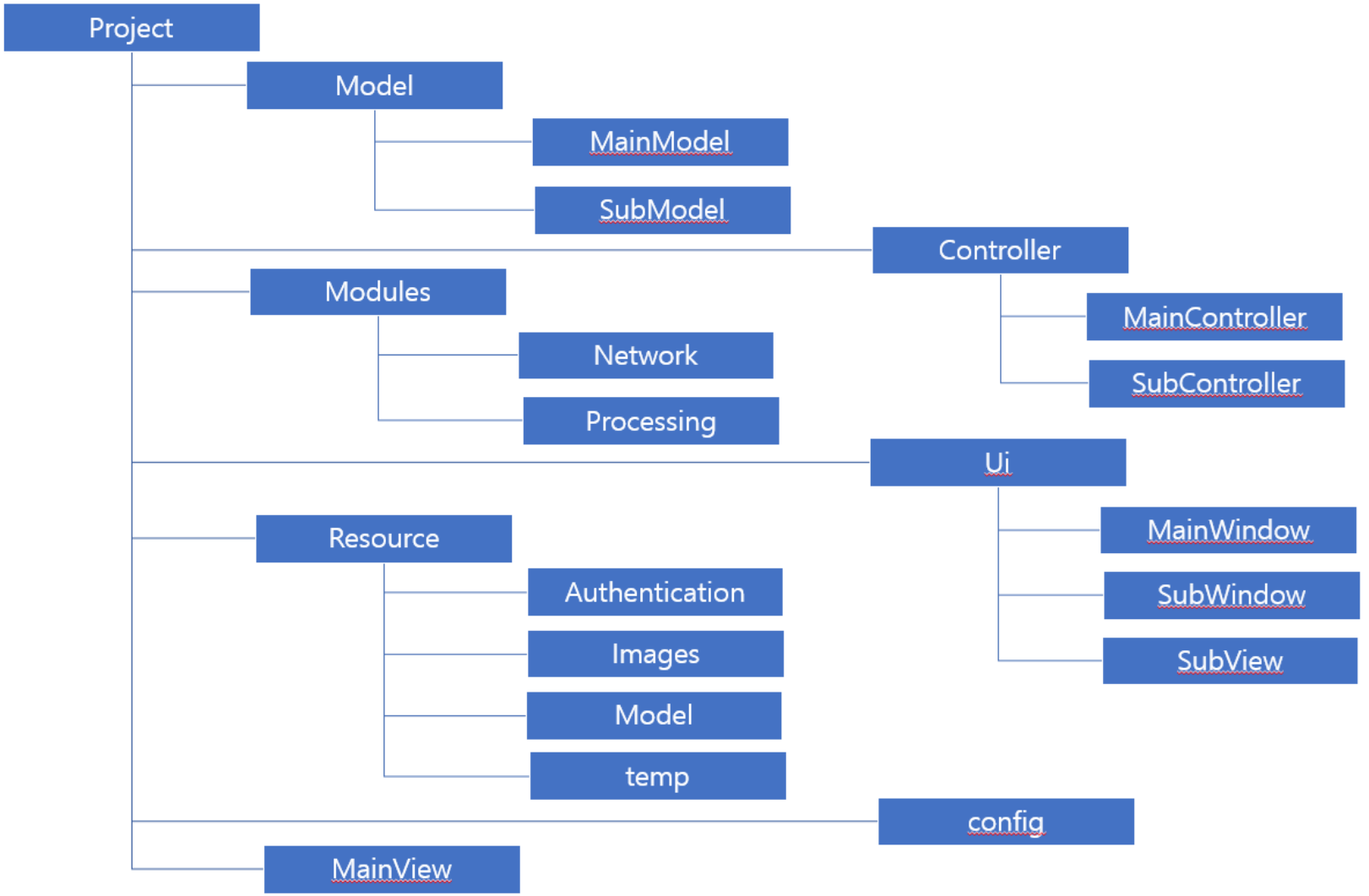


메인화면 표시

보조화면 표시

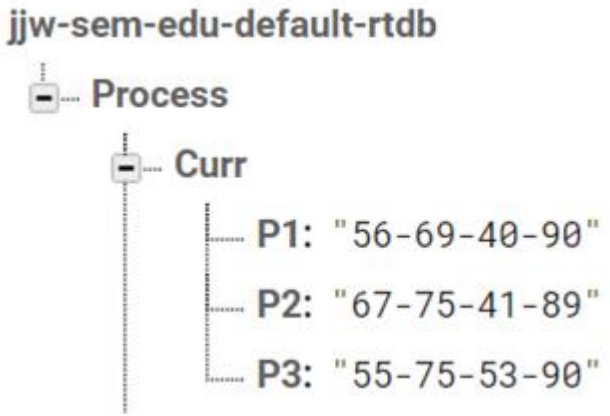
MVC : 시스템을 Model, View, Controller로 나누어 개발
다른 개발자와 협업 시 유지보수 효율이 좋음

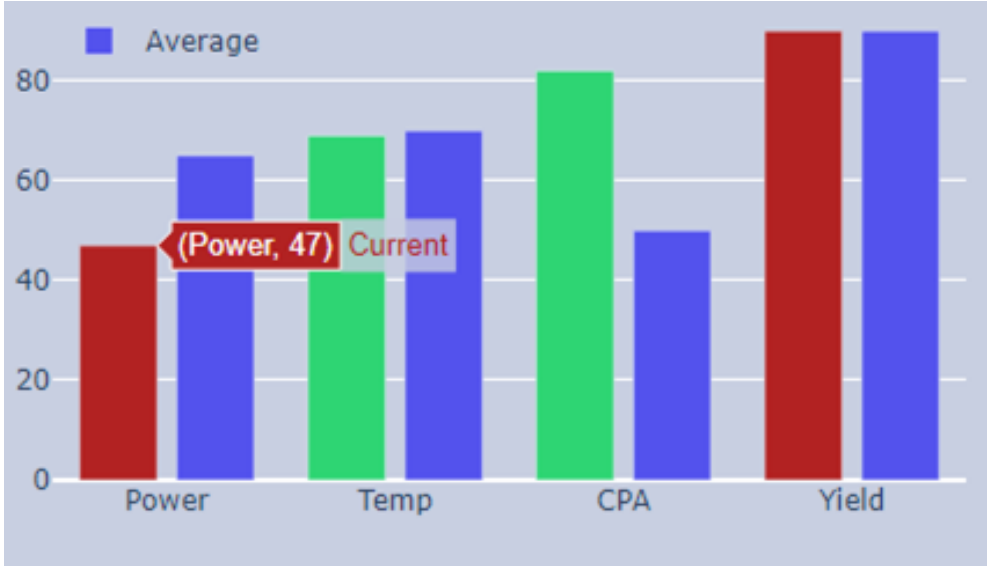
Ⅲ. 프로젝트 상세



기능별로 각각의 폴더와 파일, 모듈을 제작하여 체계적 수정, 관리 가능

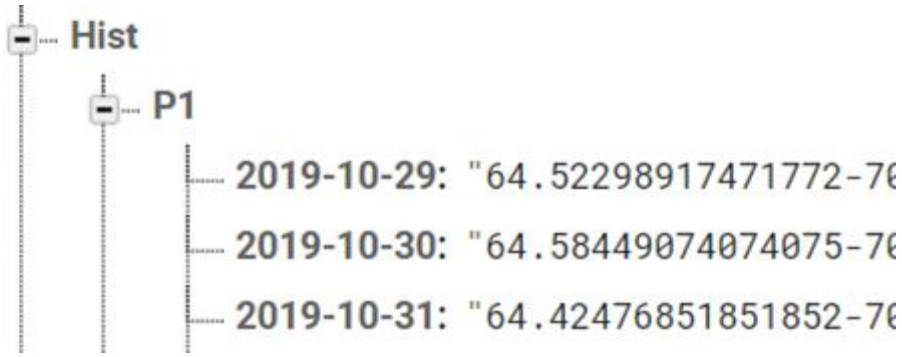
Ⅲ. 프로젝트 상세

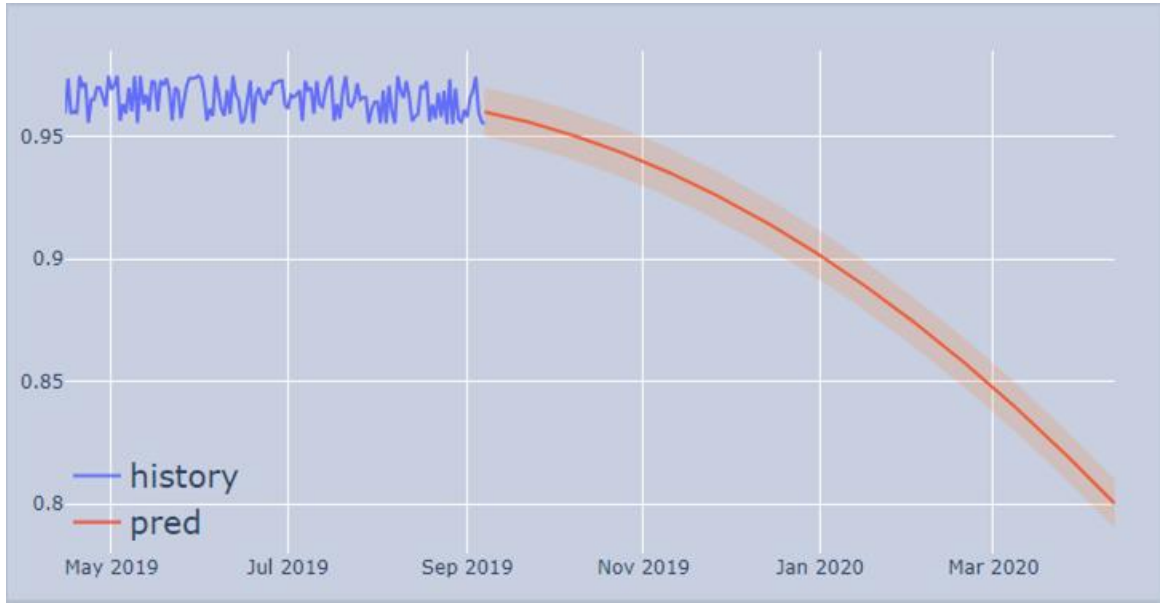




- 현재 설비의 각 값들을 기준치와 비교하여 막대그래프로 시각화
- 기준치보다 낮으면 적색,높으면 녹색으로 표현
- plotly라이브러리를 활용한 상호작용형 그래프(클릭, 드래그 등을 통해 세부 정보 확인 가능)

Ⅲ. 프로젝트 상세



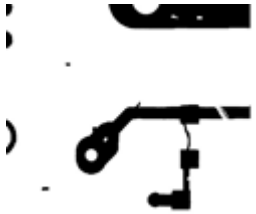


1. LSTM을 통한 접근 : 최근시점으로부터 n 시점뒤의 데이터는 예측할 수 있지만 부품교체시기를 특정하기 힘듬. -> 최근시점만 활용하기 때문
2. 가동일수, 일별 사용량의 누적 평균, 분산, 첨도와 왜도를 추가 피처로 만들고 이를 통한 머신러닝 -> 부품교체시기를 특정가능

결론 : 머신러닝(LightGBM)을 활용해 교체시기를 특정. 사이의 값은 2차회귀곡선에 오차를 추가하여 표현

Ⅲ. 프로젝트 상세

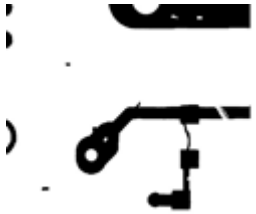
1.정상/비정상 구분



정상	비정상
0	1

시각화 어려움
->결함의 원인을 알기 힘듬

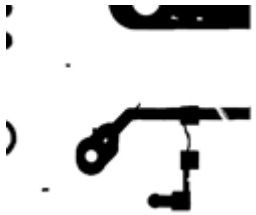
2.결함종류별 구분



단락	개방
3	1

학습이 어려움

3.결함 시각화

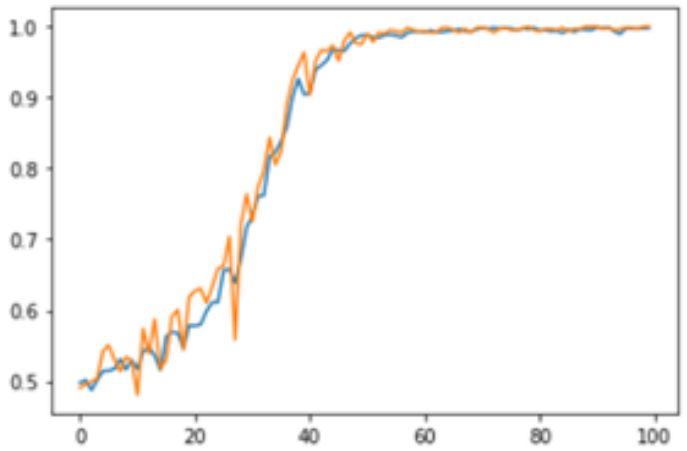


단락	개방
3	1

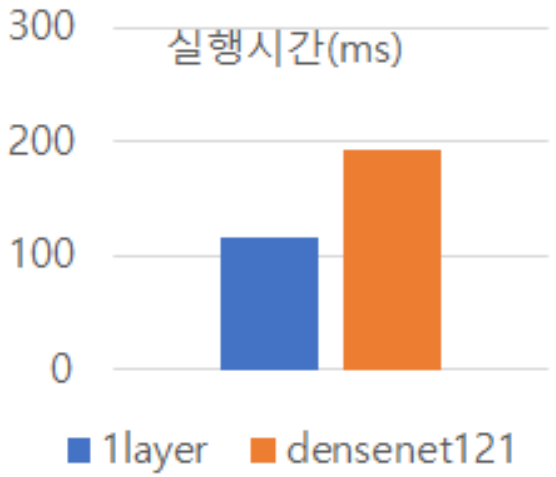
결함를 시각화도록 먼저 학습-> 결함의 종류별 개수를 찾게 학습
단계별 학습으로 학습이 수월함 & 시각화 용이

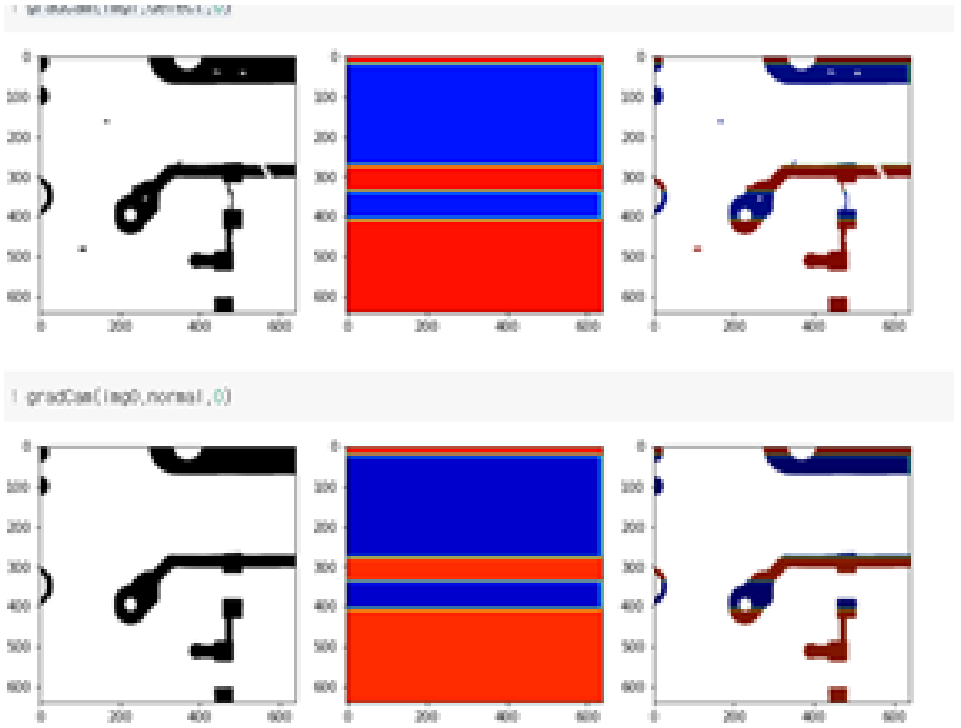
Ⅲ. 프로젝트 상세

정상/비정상 이진분류



1 layer cnn

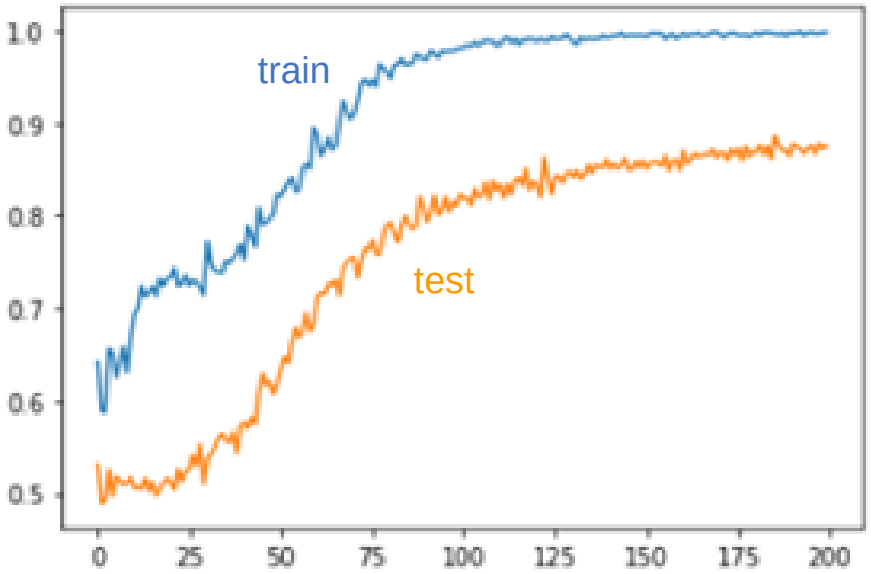




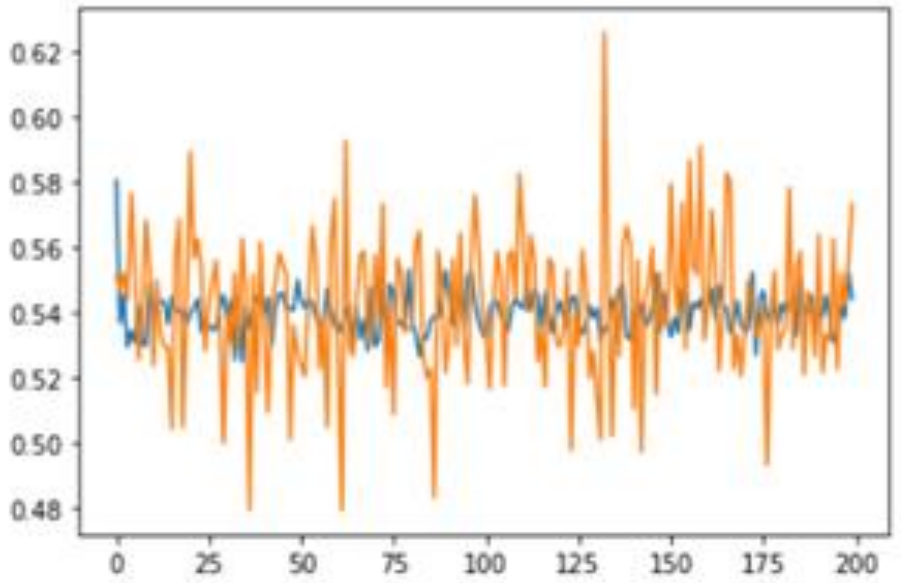
- 1Layer CNN만으로도 이진분류에서 높은 정확도와 빠른 속도
  ->깊은 망을 사용하지 않음
- 직관적이지 않은 시각화(Grad-CAM)

Ⅲ. 프로젝트 상세

결함 종류별 이진분류



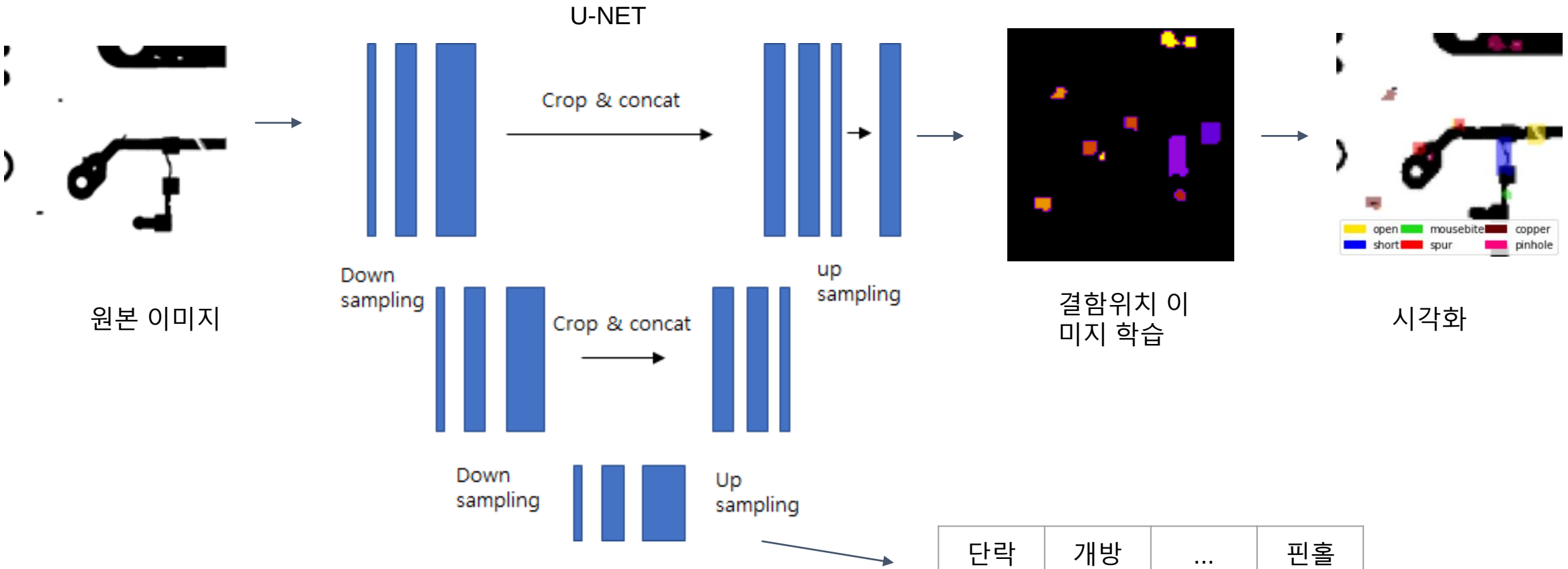
Multi Label Cross Entropy 학습
정확도



MSE
반올림한 카테고리들의 정확도 평균

오류에서 가장 큰 부분을 차지하는 종류만 감지(클래스의 불균형)
-> 특정 위치에 특정오류가 있는걸 인지할 수 있게 학습 유도 필요

Ⅲ. 프로젝트 상세



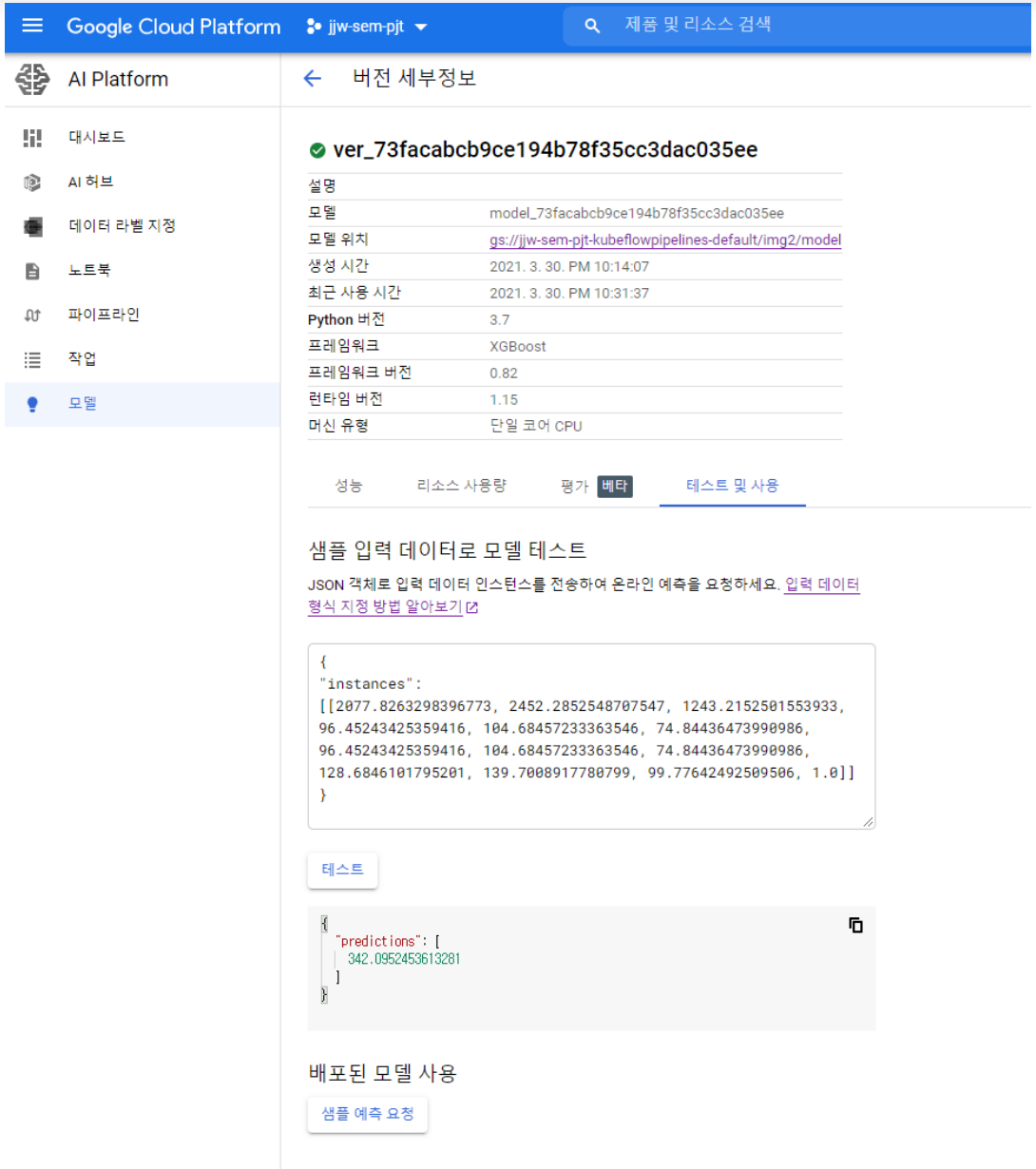
원본 이미지

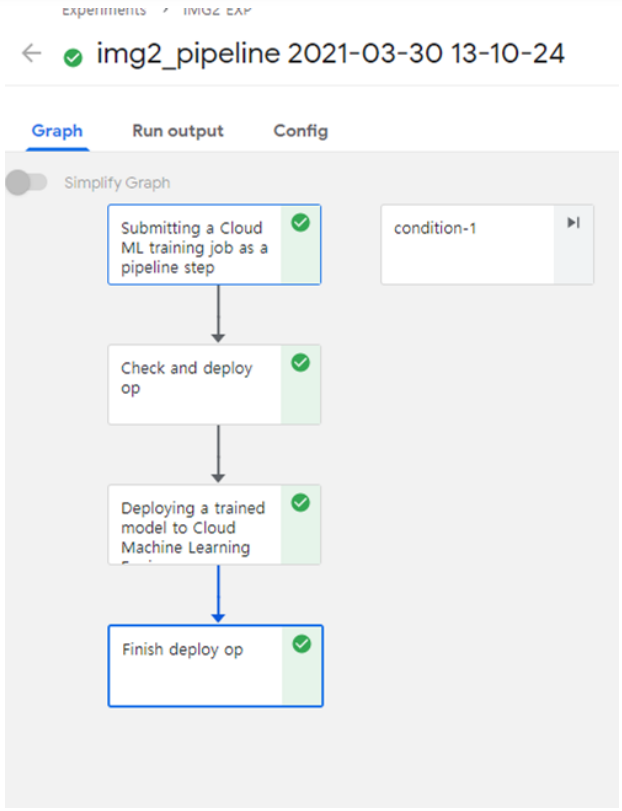
결함위치 이미지 학습

시각화

단락	개방	...	핀홀
2	1	...	2

Ⅲ. 프로젝트 상세

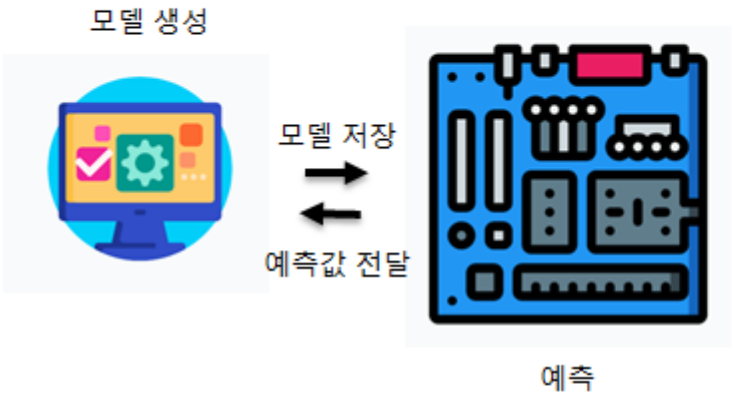




Google Cloud Platform의 Kubeflow Pipeline

- AI모델 작성 시 단계별로 분할, 협업에 용이
- 모델 갱신, 배포에 용이

AI모델 배포&예측방법 비교

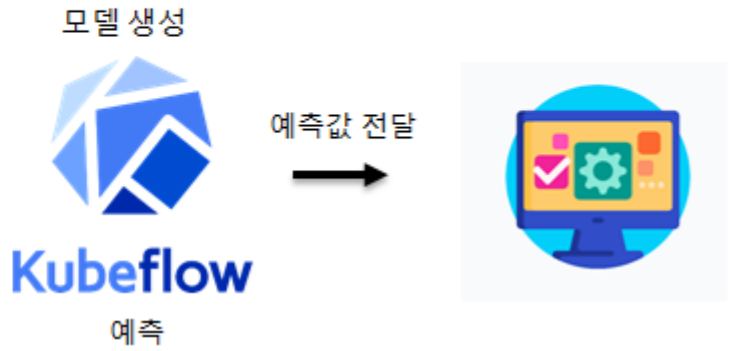


로컬 배포 & 송신부 예측

- 모델 교체가 어려움
- 송신부 제어 필요
- 송신부 H/W 비용 높음

AI 서버 배포 & 수신부 예측

- 모델 교체가 쉬움
- 서버 비용 발생



AI 서버 배포 & 예측

- 모델 교체가 쉬움
- 수신부 모델저장, 예측 불필요
- 서버 비용 높음
- 트래픽 높을 시 속도 저하
  ->병렬 연산 등의 보완 필요

Ⅳ. 프로젝트 효과

Ⅳ. 프로젝트 효과

특징
- 개발을 하기 전 설계단계에서 많은 부분을 고려(사용자, 협업, 이식성, 확장성)

기본 요구사항(설비 시각화, 수명예측)을 만족
- 머신러닝과 딥러닝을 활용한 예측, 시각화
- DB와의 실시간 통신을 구현

사용 편의성
- 상호작용형 그래프
- 단순한 조작(적/녹색으로 설비상태 표현,이미지나 그래프 클릭)

개발 편의성
- 필요시 특정 모듈이나 특정 페이지만 보수
- 유지보수과정에서 고장 최소화
- 다른 언어나 플랫폼으로 이식시 단계별로 수행 가능

개선사항
- 송신부를 구현하지 않았으므로 실제 구현시 수정 필요
- 응답속도 개선 필요