캡스톤디자인 II 중간보고서(표지)

프로젝트명 : 디지털 트윈을 활용한 스마트 팩토리 에너지 효율화 모델링 및 플랫폼 개발

캡스톤 디자인II, 중간보고서

Version 1.0

개발 팀원 명(팀리더): 박선아
서지윤
대표 연락처: 010-5177-9950
e-mail: 20222024@edu.hanbat.ac.kr

캡스톤 디자인 II 중간보고서 내용

1. 요구사항 정의서에 명세된 기능에 대하여 현재까지 진척된 결과 및 그 내용을 기술하시오.

1.1 대시보드 구현

1.1.1 소스 코드 일부 및 실행 환경

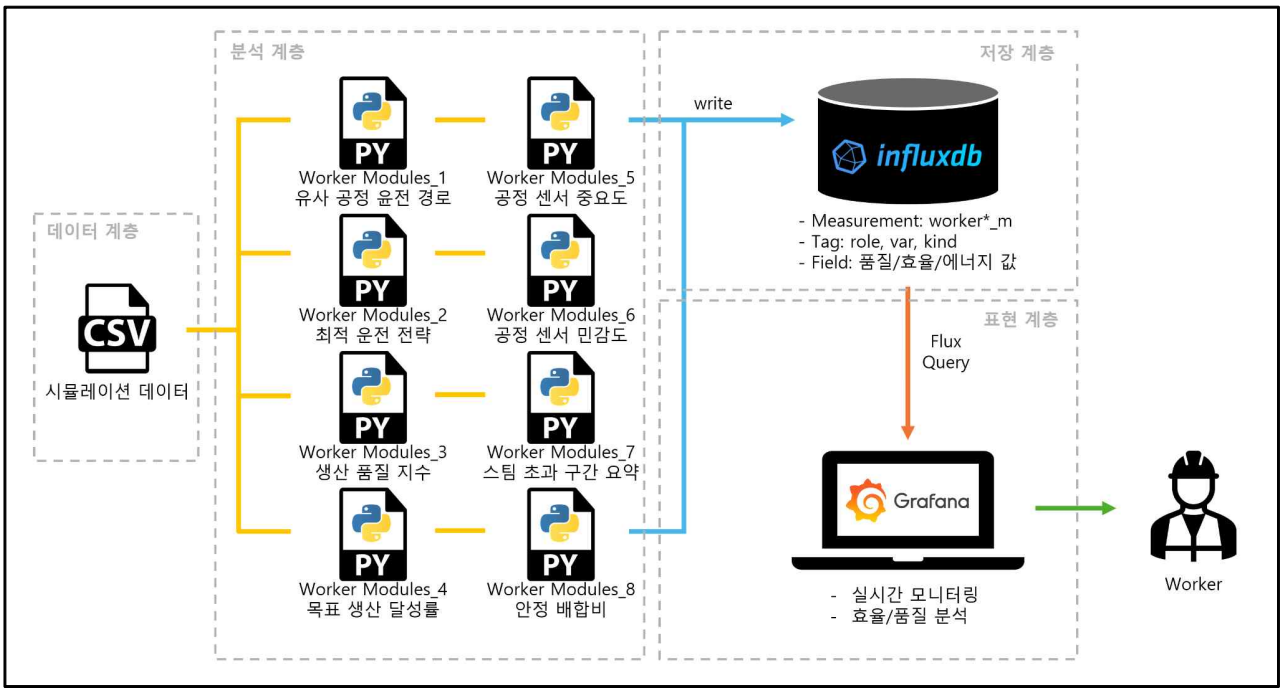
Python을 이용하여 데이터 스트리밍 및 적재 모듈을 구현하였다. 해당 코드는 1분 단위로 구성된 LOT 데이터를 초 단위로 재생할 수 있도록 변환하며, InfluxDB에 시계열 데이터로 저장한다. 또한 저장 시에는 role, kind, lot 등의 태그를 함께 부여하여 이후 Grafana 대시보드에서 다양한 조건으로 필터링 및 분석이 가능하다.

```
SECONDS_PER_MIN = 5.0
for row in df.itertuples():
    point = (
        Point("worker1_m")
        .tag("role", "current")
        .tag("kind", "value")
        .field("x1", row.x1)
        .field("x2", row.x2)
        .time(datetime.utcnow(), WritePrecision.NS)
    )
    write_api.write(bucket=INFLUX_BUCKET, record=point)
    time.sleep(SECONDS_PER_MIN)
```

환경설치 문서

- Python 3.10 이상
- 라이브러리: influxdb-client, pandas, numpy
- InfluxDB v2.7, Grafana 10.2 (ECharts 플러그인 설치)

1.1.2 시스템 구성도



<그림1. 대시보드 시스템 구성도>

1.1.3 설계 모델

본 시스템은 총 8개의 Worker 모듈로 구성되며, 각 모듈은 특정 기능(품질 지수 산출, 효율 분석, 실시간 모니터링 등)을 담당한다. 모듈에서 처리된 데이터는 InfluxDB에 저장되고 Grafana 대시보드를 통해 시각화된다.

모듈	주요 기능	입력	처리	출력 (InfluxDB)	Grafana 패널 목적
Worker_1	유사 공정 운전 경로	CSV 시뮬레이션 데이터	유사한 운전 경로 탐색 및 전처리	유사 경로 데이터가 시간순으로 기록됨	유사 경로 비교
Worker_2	최적 운전 전략	LOT 데이터	조건별 최적 전략 시뮬레이션	조건별 최적 전략 시뮬레이션	최적 전략 패널
Worker_3	생산 품질 지수	LOT 데이터	품질 지수 산출 (평균, 분산 등)	공정 운전 단위로 집계된 품질 점수	품질 지표 패널
Worker_4	목표 생산 달성률	생산 기록	목표 대비 달성률 계산	목표 대비 실제 생산 달성 정도	달성률 패널
Worker_5	공정 센서 중요도	센서 데이터	변수 중요도 분석 (기여도)	각 센서별 기여도 및 중요도 값	중요 변수 분석
Worker_6	공정 센서 민감도	센서 데이터	민감도 분석 (허용밴드 대비)	변수별 허용 범위 초과 정도와 민감도 결과	민감도 분석 패널
Worker_7	스팀 초과 구간 요약	에너지 데이터	스팀 사용량 초과 구간 탐색	스팀 사용량이 초과된 구간의 요약 정보	에너지 관리 패널
Worker_8	안정 배합비	LOT + 배합 데이터	안정적인 배합비 산출	안정적인 배합비 추천 결과	배합비 최적화 패널

<표1. 대시보드 모듈 명세서>

1.1.4 UI 분석/설계

UI는 작업자 대시보드 형태로 구현되었으며, InfluxDB에 저장된 데이터를 Grafana를 통해 실시간으로 시각화하였다. 설계 목표는 작업자가 생산 현황과 품질 상태를 직관적으로 확인하고, 유사 공정 운전 궤적적 비교와 공정 변수의 영향도를 동시에 분석할 수 있도록 하는 것이다.

<그림2. Grafana기반 대시보드 구현 UI>

- 좌측 영역: 유사 공정 운전 경로 패널
속도, 전력, 압력, 유량 등 주요 변수별로 현재 공정 운전과 과거 유사 공정 운전을 시계열 그래프로 비교한다. 작업자는 과거 운전 경로와 현재 상태를 직관적으로 비교할 수 있다.

- 상단 중앙: 목표 생산 달성률 & 스팀 사용량 예측 패널
목표 생산량 달성 정도를 색상 지표로 표현한다. 스팀 사용량을 예측 곡선으로 표시하여 에너지 효율 모니터링이 가능하다.

- 우측 상단: 생산 품질 지수 패널
현재 공정 운전 단위의 품질 지수를 시간에 따라 표시한다. 품질 안정성과 목표 품질 대비 현황을 실시간으로 확인 가능하다.

- 중앙: 최적 운전 전략 패널
공정 조건별 최적 운전 전략을 표 형태로 제시한다. 현재 값과 목표 값, 편차를 함께 제시하여 조정 방향성을 제공한다.

- 하단 좌측: 스팀 초과 구간 요약 패널
스팀 사용량이 초과된 구간을 요약하여 표시한다. 작업자가 에너지 과소비 구간을 빠르게 인지 가능하다.

- 하단 중앙: 안정 배합비 패널 (레이더 차트)
여러 원료 배합비를 레이더 차트로 시각화한다. 배합비의 안정성과 기준값과의 차이를 확

인 가능하다.

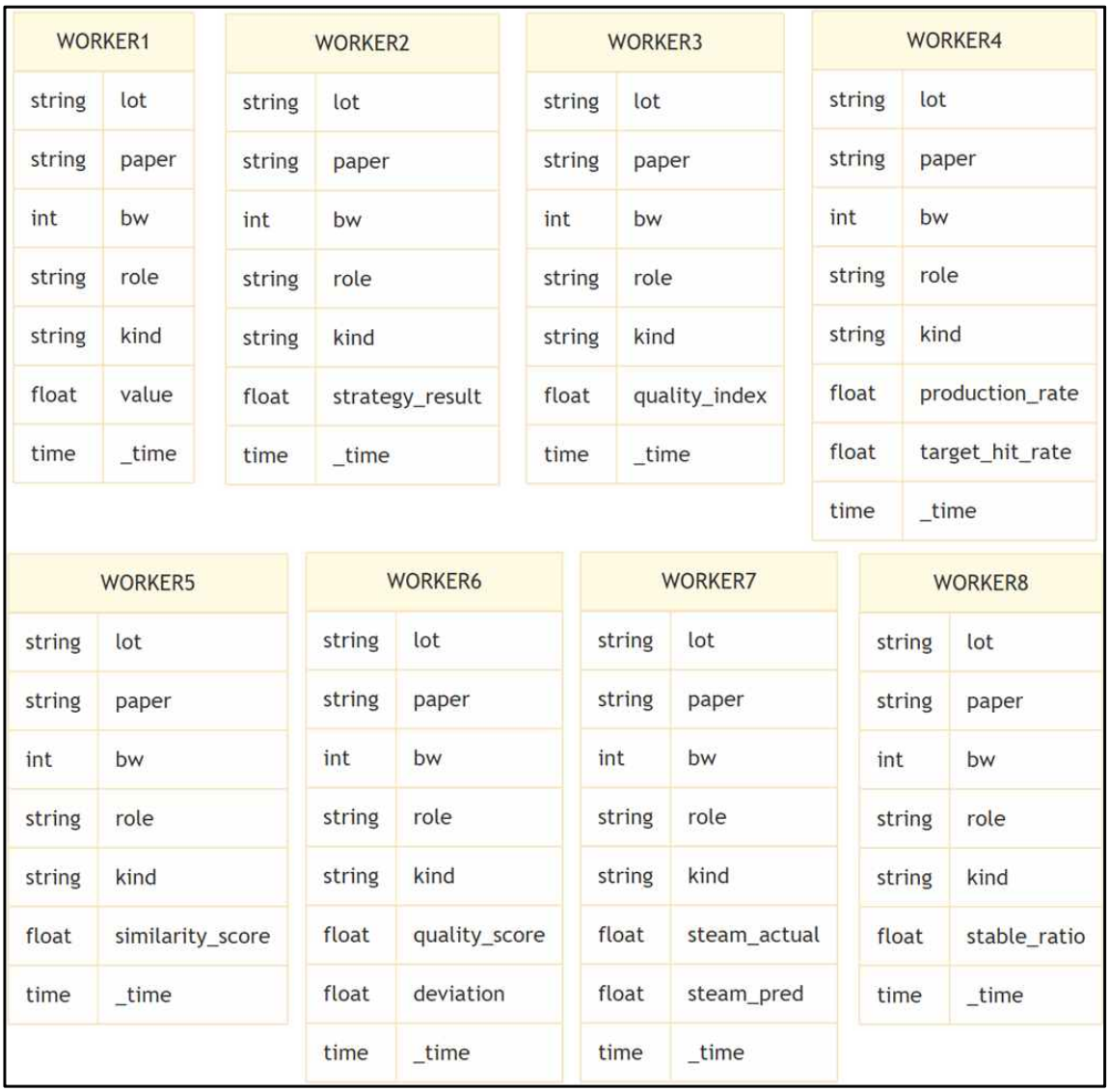
- 하단 우측: 공정 센서 중요도 & 민감도 패널
파이차트: 각 센서(변수)의 기여도와 중요도를 표시한다.
바차트: 변수별 민감도 분석 결과와 허용 범위 초과 여부를 표시한다.

1.1.5 E-R 다이어그램/DB 설계

InfluxDB 기반의 시계열 데이터베이스를 사용하여 Worker 모듈의 결과를 저장하고, Grafana 대시보드를 통해 이를 시각화하였다. InfluxDB는 전통적인 RDBMS와 달리 Measurement, Tag, Field, Time 구조를 가지므로, 이를 기준으로 DB 설계를 진행하였다.

- 데이터 저장 구조
    - Measurement: worker1_m ~ worker8_m
    - Tag: lot, paper, bw(평량), role(current/similar/history), kind(value/band/summary), var(x1~x5 등)
    - Field: 품질 지수, 효율 값, 목표 달성률, 스팀 사용량, 배합비, 중요도/민감도 지표 등
    - Time: 시계열 데이터의 기준 시간 (자동 생성)

- Worker 모듈별 저장 스키마
각 Worker 모듈은 독립적인 Measurement로 저장되며, 공통적으로 lot, paper, bw, role, kind와 같은 태그를 가지며, 모듈별 특화된 필드를 저장한다.
    - Worker1: 유사 운전 경로 값
    - Worker2: 최적 운전 전략 결과
    - Worker3: 생산 품질 지수
    - Worker4: 생산 달성률 및 목표 달성 여부
    - Worker5: 유사 LOT 유사도
    - Worker6: 품질 점수 및 편차 값
    - Worker7: 스팀 사용량(실측/예측)
    - Worker8: 안정적인 배합비 값



<그림3. 기반 대시보드 DB 스키마 설계>

1.1.6 테스트 계획/테스트 케이스

대시보드 테스트는 요구사항 정의서에 기재된 기능 요구사항을 기준으로 진행되었다. Python 스트리밍 모듈, InfluxDB, Grafana 대시보드 환경에서 기능별 검증을 수행하였으며, 각 항목은 요구사항 고유번호(TER-001 ~ TER-003)에 따라 결과를 정리하였다.

- TER-001 데이터 스트리밍 및 적재 테스트

- 테스트 방법

Python 스트리밍 모듈(worker2_m)을 실행하고 SECONDS_PER_MIN = 5로 설정하여 1분 단위 데이터를 5초 간격으로 전송하였다. InfluxDB에 데이터가 누락 없이 기록되는지와 태그(role=current)가 정상적으로 저장되는지를 확인하였다.

- 테스트 결과

[그림4]는 스트리밍 모듈 실행 시 출력된 로그 화면이다. cutoff_min 값이 증가함에 따라 5포인트씩 role=current 태그로 데이터가 기록되는 것을 확인하였다. 또한 InfluxDB Data Explorer에서도 _time 값이 순차적으로 증가하며 새로운 데이터가 누적되는 것을

확인하였다([그림5]).

```
18]  14.1s

..   [1/54] cutoff_min=1 → 5 pts (role=current @ now)
     [2/54] cutoff_min=2 → 5 pts (role=current @ now)
     [3/54] cutoff_min=3 → 5 pts (role=current @ now)
```

<그림4. Python 스트리밍 로그>

- TER-002 품질, 효율 점수 계산 정확성 테스트

- 테스트 방법

Worker3, Worker4, Worker6 모듈을 실행하여 LOT A3239, B3558을 대상으로 품질 지수(score), 생산 달성률, 품질 점수 및 편차를 계산하였다. InfluxDB에 기록된 값과 엑셀로 계산한 기준값을 비교하여 ±0.5 이내의 오차 범위 내에 있는지를 검증하였다.

- 테스트 결과

[그림4]는 InfluxDB Data Explorer에서 조회한 worker3_m 모듈의 품질 점수 기록 화면이다. LOT A3239의 score 값이 78~89 범위로 나타났으며, 이는 사전 계산값과 일치하는 수준이었다.

table	_measurement	_field	_value	_start	_stop	_time	lot	role
0	worker3_m	score	79.66057442905608	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T10:58:30.000Z	B3558	current
0	worker3_m	score	78.80949206865023	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:02:20.000Z	B3558	current
0	worker3_m	score	81.21199764281253	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:06:10.000Z	B3558	current
0	worker3_m	score	83.29255359051956	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:10:00.000Z	B3558	current
0	worker3_m	score	83.07403078758146	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:13:50.000Z	B3558	current
0	worker3_m	score	85.58586300323954	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:17:40.000Z	B3558	current
0	worker3_m	score	84.79538044124007	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:21:30.000Z	B3558	current
0	worker3_m	score	84.76859672995036	2025-10-01T15:00:00.000Z	2025-10-02T14:00:00.000Z	2025-10-02T11:25:20.000Z	B3558	current

<그림5. InfluxDB Data Explorer-품질 점수 기록>

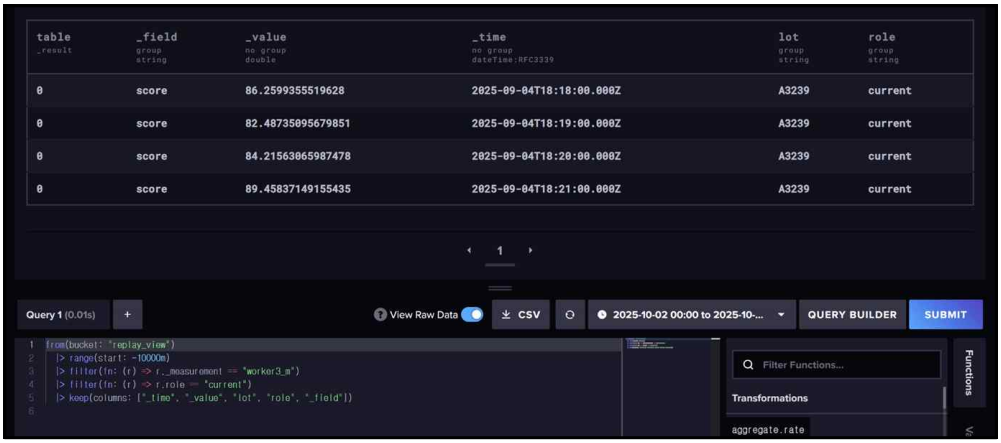
- TER-003 작업자 대시보드 UI 반응 테스트

- 테스트 방법

Grafana 대시보드에서 Refresh Interval을 5초로 설정하고, Worker 모듈 실행 후 패널들이 주기적으로 갱신되는지 확인하였다. 특히 목표 생산 달성률, 품질 지수, 스팀 사용량 예측, 유사 공정 운전 비교, 센서 중요도, 민감도 패널을 중심으로 실시간 반응을 점검하였다.

- 테스트 결과

[그림6]은 실제 작업자 대시보드 화면이다. 품질 지수, 목표 생산 달성률, 스팀 사용량 예측 그래프가 실행 직후부터 5초 간격으로 실시간 갱신되었으며, 유사 공정 운전 비교 패널과 센서 분석 차트 역시 정상적으로 데이터가 반영되었다.



< 그림6. Grafana 작업자 대시보드-실시간 패널 반응>

1.2 디지털 트윈 구현

1.2.1 소스 코드 일부 및 실행 환경

Python을 이용하여 기존 캡스톤 디자인 I 의 모델링 코드를 기반으로 한 스팀 사용량 예측 및 품질 점수 산출 모듈을 구현하였다. 해당 모듈은 LSTM 모델을 통해 LOT 데이터를 입력받아 예측 스팀량과 품질 지수를 계산하며, 결과는 Flask API를 통해 외부에서 호출할 수 있다. 예측에 필요한 공정 데이터는 influxDB에 저장된 시계열 데이터를 불러와 전처리 및 학습 데이터로 활용하였다. 예측 결과는 Flask API 응답을 통해 Unreal Engine으로 전달된다. Unreal Engine에서는 Varest 플러그인을 통해 Flask API를 호출하고, 응답된 데이터를 UMG 위젯으로 시각화하여 작업자가 직관적으로 확인할 수 있도록 하였다.

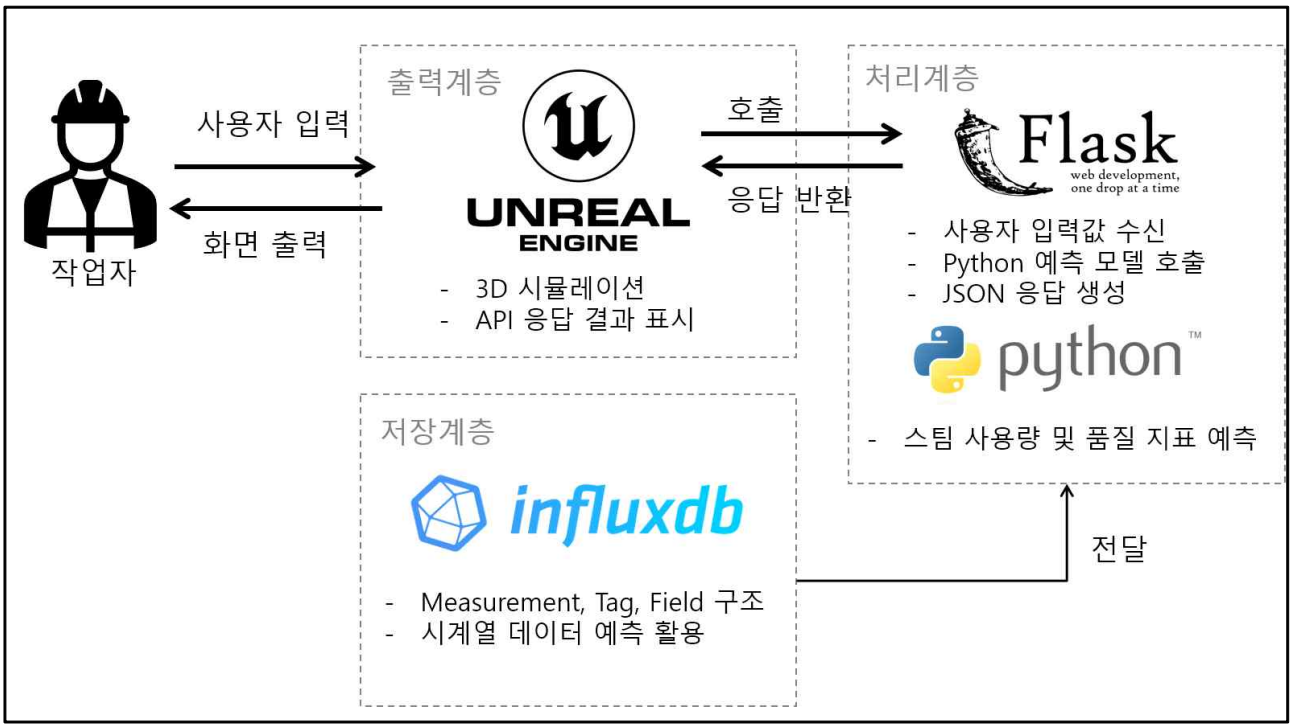
```
# app.py
from flask import Flask, request, jsonify
from quality_preprocess import calculate_current_lot_score_api
from utils.preprocess import load_resources, predict_steam

app = Flask(__name__)

# 리소스 캐싱
resources = load_resources()

@app.route("/predict/steam", methods=["POST"])
def predict_steam_api():
    data = request.get_json()
    lot = data.get("lot")
    minutes = data.get("minutes", 24)
    result = predict_steam(lot, minutes, resources)
```

return jsonify(result)

환경설치 문서

- Python 3.10 이상
- 라이브러리: influxdb-client, pandas, numpy, scikit-learn, flask
- InfluxDB v2.7, Unreal Engine 5.3 (VaRest 플러그인 설치)

1.2.2 시스템 구성도



<그림7. 디지털 트윈 시스템 구성도>

1.2.3 설계 모델

본 시스템은 Flask API 모듈, Python 예측 모델 모듈, Unreal Engine 모듈로 구성되며, 각 모듈은 입력값 처리, 예측 수행, 결과 반환 및 시각화 기능을 담당한다. Flask API는 입력값을 받아 Python 예측 모델을 호출하여 스팀 사용량 및 품질 지표를 계산하고, JSON 형식으로 결과를 반환한다. Python 예측 모델은 InfluxDB에 저장된 시계열 데이터를 기반으로 학습된 LSTM 모델을 활용하여 입력된 LOT과 시간 조건에 따른 예측 스팀량과 품질 점수를 산출한다. Unreal Engine 모듈은 작업자로부터 입력값을 받아 Flask API를 호출하며, 응답받은 예측 결과를 UI 화면에 출력한다.

모듈	주요 기능	입력	처리	출력
Flask API	사용자 입력 처리 및 예측 요청 관리	LOT, 시간	입력값 수신 → Python 예측 모델 호출 → JSON 생성	예측 결과 JSON 응답
Python 모델	스팀 사용량 및 품질 지표 예측	LOT, 시간	InfluxDB 기반 데이터 활용, LSTM 기반 예측 수행	예측 스팀량, 품질 지표
Unreal Engine	사용자 입력 및 결과 시각화	LOT, 시간 (UI 입력)	LOT/시간 값 입력, API 호출, 응답 결과 표시	화면(UI) 출력

<표2. 디지털 트윈 모듈 명세서>

1.2.4 UI 프로토타입 분석/설계

UI는 Unreal Engine 기반의 디지털 트윈 환경과 UMG 위젯 인터페이스 형태로 구현되었다. 설계 목표는 작업자가 가상 3D 환경에서 공정 조건을 입력하고, Flask API와 연동하여 Python 예측 모델 결과를 실시간으로 확인할 수 있도록 하는 것이다.

<그림8. Unreal Engine 기반 디지털 트윈 시뮬레이션 환경>

<그림9. UMG 위젯 기반 입력/출력 UI>

- [그림 8] 3D 시뮬레이션 화면

Unreal Engine 에셋을 활용해 공정 라인을 가상 공간에 배치하였다. 실제 공정을 간략하게 구현한 3D 공간에서 작업자는 시뮬레이션을 진행할 수 있다.

- [그림 9] 좌측: UMG 위젯 입력 패널

LOT, 시간을 입력할 수 있는 텍스트 창과 확인 버튼을 제공한다. 사용자가 직접 조건을 입력하고 분석을 요청하면 이를 수행한다. 입력이 잘못되었을 경우, 안내 문구를 출력한다.

- [그림 9] 우측: UMG 위젯 출력 패널

사용자가 입력한 LOT, 시간에 대한 Flask API 응답(JSON)을 기반으로 예측 결과를 표시한다. 품질 점수, 스팀 사용량, 제어변수 초과 여부 등의 결과를 텍스트 형태로 출력한다.

1.2.5 E-R 다이어그램/DB 설계

디지털 트윈 시뮬레이션시 InfluxDB 기반 시계열 데이터베이스를 활용하여 예측에 필요한 공정 데이터를 불러오고, 전처리 및 학습 데이터로 활용하였다. 예측 결과(스팀 사용량, 품질 점수)는 Flask API를 통해 Unreal UI에 직접 표시되며, 본 DB 설계는 예측 입력 데이터 활용에 중점을 두었다.

- 데이터 저장 구조
    - Measurement: paper_data
    - Tag: lot, paper, bw(평량), season
    - Field: 주요 공정 변수(x1~x20 등), 계산 스팀 사용량, 품질 관련 지표
    - Time: 시계열 데이터의 기준 시간 (자동 생성)

1.2.6 테스트 계획/테스트 케이스

디지털 트윈 시뮬레이션 테스트는 요구사항 정의서에 기재된 기능 요구사항을 기준으로 진행되었으며, Flask API와 Unreal Engine UMG UI 연동 과정을 검증하였다.

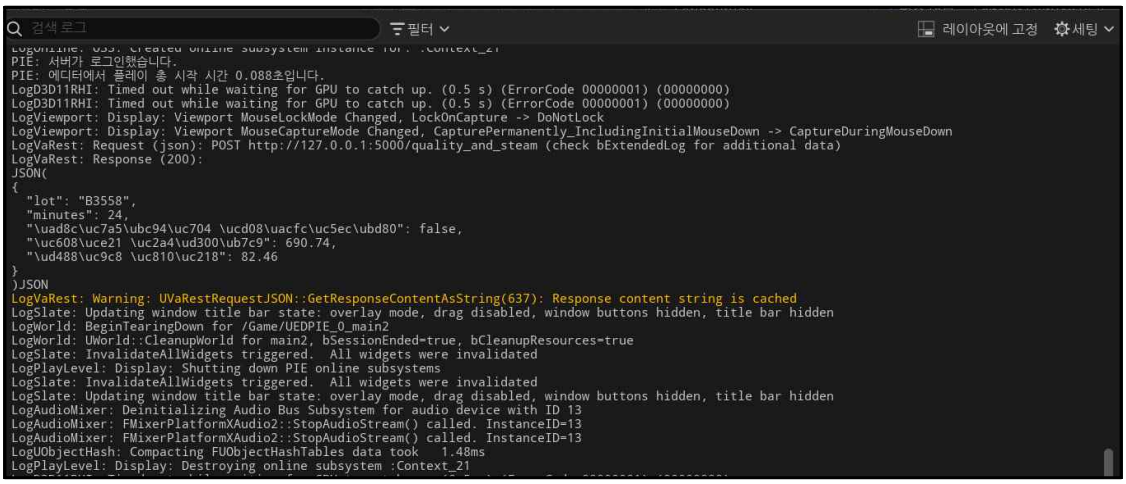
- TER-004 디지털 트윈 시뮬레이션 연동 테스트
    - 테스트 방법

Unreal Engine 시뮬레이션을 실행하고, UMG UI 입력값을 통해 사용자 요청사항을 전송해 API 응답 표시 과정을 검증하였다. Flask API 호출 시 입력값(LOT, 시간)에 따른 예측 결과(JSON)가 정상적으로 반환되는지, 화면에 손실 없이 표시되는지, API 응답 후 전체 출력 흐름이 평균 5초 이내에 완료되는지를 확인하였다.

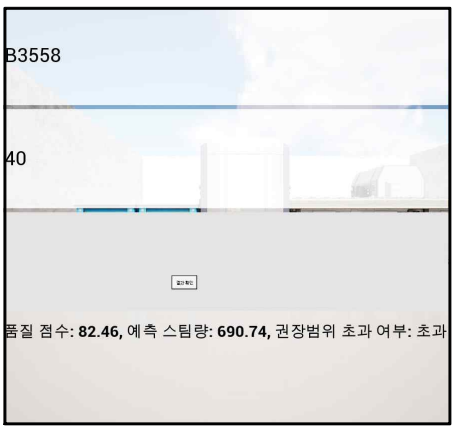
    - 테스트 결과

Unreal Engine을 통해 디지털 트윈 시뮬레이션 실행 중 Flask API를 호출한 결과, 평균 응답 시간은 5초 이내(67ms), 예측 결과(JSON) 정상적 반환, 응답 코드 200으로 정상 처리됨을 확인하였다([그림10]). 또한 [그림 11]을 통해 시뮬레이션 실행 및 종료 로그가 정상 기록되어 시스템이 정상적으로 동작함을 확인할 수 있고, [그림 12]를 통해 화면에도 손실 없이 표시되는 것을 확인할 수 있었다.

<그림10. Python 터미널 로그>



<그림11. Unreal Engine 출력 로그>



<그림12. Unreal Engine UMG 위젯 출력 결과>

2. 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

2.1 대시보드 구현

- 실시간 데이터 가시성 확보

대시보드 구현 초기에는 Grafana의 기본 새로고침 주기(Refresh Interval)가 1분으로 설정되어 있어, 데이터 갱신 속도가 더뎌 사용자 경험 측면에서 한계가 있었습니다. 이를 개선하기 위해 Python 데이터 적재 모듈을 수정하여 1분 단위가 아닌 5초 주기로 InfluxDB에 데이터를 업데이트하도록 조정했습니다. 동시에 Grafana의 Refresh Interval을 동일하게 5초로 단축하고, Flux 쿼리에는 aggregateWindow 함수를 적용해 5초 단위 집계가 가능하도록 최적화했습니다. 이러한 개선을 통해 대시보드의 응답성이 크게 향상되었으며, 작업자가 데이터를 지연 없이 확인할 수 있는 환경을 구축함으로써 요구된 모니터링 기능을 안정적으로 구현할 수 있었습니다.

- 대시보드 가독성 및 직관성 개선

초기 대시보드는 보여주고자 하는 지표가 과도하게 많아 화면이 복잡해지고, 일부 패널은 작업자 입장에서 실질적인 의미가 부족했습니다. 이로 인해 중요한 정보가 묻히고, 대시보드를 통해 즉각적인 상황 판단을 내리기 어려운 문제가 있었습니다.

이를 개선하기 위해 작업자가 실제로 참고해야 할 지표를 중심으로 화면을 재구성하고, 의

미 전달력이 낮은 패널들은 정리했습니다. 또한 관련된 지표들을 묶어 하나의 영역에서 시각화함으로써, 동일 맥락의 데이터를 비교·이해하기 쉽게 배치했습니다. 여기에 색상 대비, 그래프 유형, 배치 구조 등을 전반적으로 조정하여 가독성과 직관성을 강화했습니다.

그 결과, 작업자는 대시보드에서 핵심 정보를 빠르게 파악하고 상황에 맞는 판단을 내릴 수 있는 환경을 확보할 수 있었으며, 전체적인 모니터링 효율성과 사용 편의성이 향상되었습니다.

- Grafana ECharts 플러그인 문제 해결

대시보드 기능을 확장하기 위해 레이더 차트를 구현하고자 했습니다. 하지만 Grafana의 기본 차트 유형에는 레이더 차트가 포함되어 있지 않아, 별도의 ECharts 플러그인을 설치해야 했습니다. 문제는 이 플러그인이 일반적인 시각화 설정 방식과 달리 코드를 직접 작성해야 한다는 점이었습니다. 예상보다 복잡한 과정이었고, 처음에는 차트가 올바르게 표시되지 않거나 오류가 반복되어 당황했습니다. 특히 속성 정의 방식에 익숙하지 않아 원인 파악에 많은 시간이 소요되었습니다. 하지만 포기하지 않고 공식 문서와 커뮤니티 자료, GitHub 이슈를 계속 참고하면서 문제를 하나씩 해결했습니다. 그 과정에서 단순히 레이더 차트를 구현하는 것에 그치지 않고, 예상치 못한 기술적 문제를 마주했을 때 스스로 학습하고 해결책을 찾아내는 능력을 키울 수 있었습니다. 또한 이 경험을 통해 프로젝트를 진행할 때는 기술적 리스크를 고려해 일정에 충분한 여유를 두는 것이 중요하다는 점을 깨닫게 되었습니다.

- Unreal Engine과 Grafana 대시보드 연동 방안

Unreal Engine과 Grafana 대시보드의 연동 핵심은 사용자가 3D 환경에서 특정 설비(asset)를 선택했을 때, 대시보드의 관련 패널이 해당 설비의 상세 지표를 즉시 보여주는 것입니다. 이를 구현하기 위한 방안으로 게이트웨이 서버를 활용하는 방식이 가장 적합하다고 판단했습니다. Unreal Engine에서 사용자가 선택한 asset_id를 게이트웨이 서버로 전송하면, 서버는 이를 React 컨테이너에 전달하여 Grafana 임베드 URL의 var-asset_id 변수를 갱신합니다. 이 과정을 통해 대시보드 패널이 즉시 필터링되어, 사용자가 3D 모델과 세부 데이터를 자연스럽게 오가며 확인할 수 있습니다.

대시보드 자체의 가독성을 높이기 위해 관련 패널들을 그룹화하고, 선택된 설비에는 하이라이트 효과를 적용할 예정입니다. 또한 기존 InfluxDB의 태그 체계(role, var, kind 등)를 그대로 활용하여 동적 쿼리 필터링을 구성함으로써, 시스템 복잡도를 줄이면서도 높은 상호작용성을 확보할 수 있습니다. 이후 실제 구현 과정에서 단계적으로 적용해 나갈 계획입니다.

2.2 디지털 트윈 구현

- 디지털 트윈 구현 방식 변경

초기에 저희의 디지털 트윈 구현 계획은 실제 공정을 디지털로 구현하고 물리모델과 직접 연결하는 것이었습니다. 이를 위해 Unreal Engine 기반으로 공정을 모델링하고, 수식을 적용하여 데이터 입력 시 결과가 출력되도록 하는 단순 시뮬레이션 구조를 구상했습니다. 그러나 실제 모델링이 단순 수식으로 표현하기에는 복잡했고, 전체 구조와도 잘 맞지

않았습니다. 이 과정에서 DB 구축 및 API 연결을 새롭게 시도하게 되면서, 기존 “언리얼 → 수식” 흐름이 “언리얼 → API 연결 → Python 모델 실행 + DB 구축 및 연동”으로 크게 변경되었습니다.

이 과정에서 Python 코드, Flask API, Unreal Engine 블루프린트 등 많은 부분을 중간에 수정해야 했으며, 이미 진행된 내용을 처음부터 다시 정의하고 진행하는 것은 어려웠으며, 많은 시간이 소요되었습니다. 그러나 모델링 및 초기 구현 시 이런 부분을 미리 고려하지 못한 것은 제 준비 부족이라고 판단했고, 여러 번의 시행착오 끝에 방향을 잡아 결과적으로 전체적인 시스템 아키텍처를 더 현실적인 형태로 보완할 수 있었습니다. 이를 통해 단순히 기능을 구현하는 것에 집중하기보다, 프로젝트 초기에 전체적인 구조를 정의하고 시작하는 것, 유연한 구조를 계획하는 것이 중요하다는 것을 깨달았습니다.

- Unreal Engine 블루프린트 연결 과정의 문제와 해결

Unreal Engine에서 UI 입력, 에셋 동작, 출력값을 연결하기 위해서는 블루프린트 노드들을 하나하나 연결해야 했습니다. 그러나 Unreal Engine을 처음 사용하다 보니 노드들의 실행선과 타깃 선을 어디에 어떻게 연결해야 할지 알기 어려웠고, 버전 변경으로 인해 노드 이름, 구조, 동작 방식이 달라지는 경우도 많았습니다. 이로 인해 다 모든 노드를 다 연결한 뒤 실행을 해도 동작이 제대로 되지 않거나, 흐름이 꼬이는 문제가 반복되었습니다.

이를 해결하기 위해 공식 문서와 커뮤니티의 자료를 참고하면서, 현재 구현한 흐름을 그림으로 정리해 보았습니다. 각 파트별로 주요 노드와 연결 순서를 명확히 정의하고, 전체적인 흐름도를 완성한 뒤, 이 흐름도를 기준으로 차근차근 진행하니 문제가 줄어들었습니다. 또한, 수정 사항이 발생했을 때도 어떤 파트를 고쳐야 하는지가 분명해져, 보다 안정적으로 구현을 이어갈 수 있었습니다. 이 과정을 통해 단순히 블루프린트를 연결하는 기술뿐만 아니라, 복잡한 시스템 흐름을 구조화하고 문제 해결을 체계화하는 방법을 배울 수 있었습니다.

- 프로젝트 일정 지연과 개선 방향

당초 계획은 디지털 트윈 구현과 시뮬레이션을 캡스톤 중간발표 시점까지 완성하는 것이었습니다. 그러나 구현 방식 변경, 언리얼 블루프린트 시행착오, API 및 DB 연동 과정에서 예상보다 많은 시간이 소요되었습니다. 그 결과 현재는 기본적인 구조와 연결 기능은 구현했지만, 다양한 기능적 요소를 포함한 완성 단계까지는 도달하지 못했습니다.

남은 기간에는 현재 구현된 기본 구조를 바탕으로 디지털 트윈 공정의 세부 구현을 확장하고, API 연결 범위를 넓혀 더 다양한 데이터와의 연동이 가능하도록 보완할 계획입니다. 이를 통해 보다 다양한 조건 변화와 시뮬레이션을 진행할 수 있도록 하여 단순한 결과 확인을 넘어, 공정 최적화에 도움이 되는 디지털 트윈을 구현하고자 합니다. 이번 경험을 통해, 사용해 보지 않은 기술 학습에 필요한 시간, 시행착오 해결 시간, 변경 가능성 및 위험 요소를 미리 고려하여 여유 있는 일정을 계획하는 것이 중요하다는 점을 배울 수 있었습니다.

[별첨1]

프로젝트명 : 디지털 트윈을 활용한 스마트 팩토리 에너지 효율화 모델링 및 플랫폼 개발

소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더):박선아
서지윤
대표 연락처: 010-5177-9950
e-mail: 20222024@edu.hanbat.ac.kr

목차

1. 시스템 개요

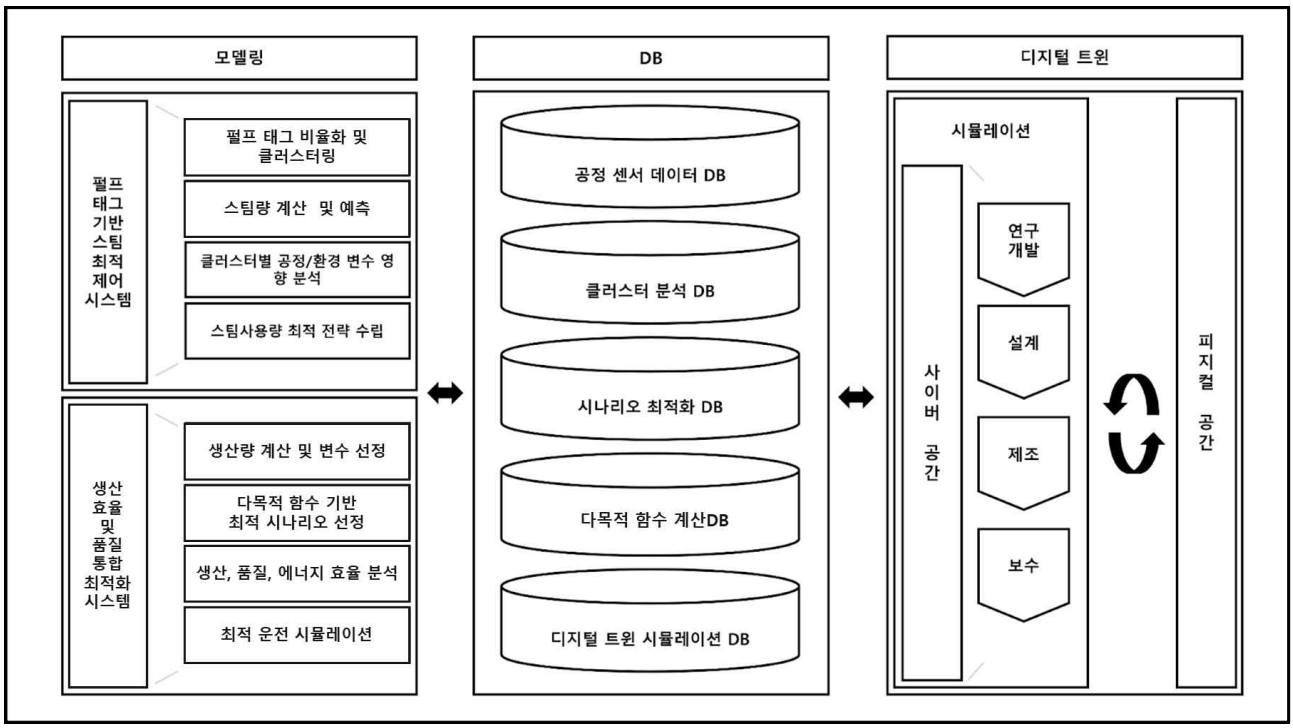
1.1 사업 목표

본 프로젝트의 사업 목표는 제지 공정 데이터를 활용하여 품질 및 효율 분석 플랫폼을 구축하는 것이다. 이를 위해 InfluxDB를 활용한 시계열 데이터 수집 및 저장 환경을 마련하였고, Python을 기반으로 한 다목적 함수 분석 기법을 적용하여 품질, 생산, 에너지 효율을 하나의 통합 지표로 평가할 수 있도록 하였다. 또한 Grafana 및 ECharts를 활용하여 작업자가 직관적으로 데이터를 확인할 수 있는 대시보드 시각화 환경을 구현하는 것을 목표로 한다. 이와 동시에 Flask API를 활용하여 Python 기반 예측 모델과 Unreal Engine을 연동함으로써, 작업자가 공정 데이터를 시뮬레이션할 수 있는 디지털 트윈 환경을 구현하는 것을 또 하나의 목표로 한다.

1.2 추진 범위

추진 범위는 데이터 파이프라인 구축, 분석 기능 구현, 그리고 시각화 환경 개발, 디지털 트윈 시뮬레이션 환경 구축으로 나눌 수 있다. 데이터 파이프라인은 제지 공정에서 발생하는 데이터를 스트리밍 및 재생 방식으로 시계열 데이터베이스(InfluxDB)에 저장할 수 있도록 구성하였다. 분석 기능은 기준값과 허용 범위를 활용하여 품질 점수를 산출하고, 생산 이력과 비교하여 유사한 LOT을 추천하며, 각 공정 변수가 결과에 미치는 영향을 정량적으로 평가할 수 있도록 하였다. 시각화 환경은 Grafana 대시보드를 기반으로 하여 작업자가 실시간으로 품질 점수와 생산 효율을 확인할 수 있으며, 계산값과 실제값의 차이, 변수별 기여도 등의 정보를 직관적으로 파악할 수 있도록 설계하였다. 디지털 트윈 시뮬레이션 환경은 Flask API를 통해 모델 예측 결과와 품질 지표를 Unreal Engine에 연동하고, UMG 기반 UI를 통해 가상 3D 공간에서 시각화하여 작업자가 시뮬레이션을 기반으로 검증을 수행할 수 있도록 하였다.

1.3 시스템 구성도



<시스템 구성도>

2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001
요구사항 명칭		데이터 스트리밍 및 분석 모듈
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	제지 공정 데이터를 스트리밍 및 재생 방식으로 처리하고, 품질, 생산, 에너지 효율을 계산하는 Python 기반 모듈
	세부 내용	- 장비 품목 : Python 실행 환경 - 장비 수량 : 1식 - 장비 기능 : 공정 데이터 재생, AI 모델 학습 및 데이터 분석 - 장비 성능 및 특징 : 시계열 데이터 연속 처리, InfluxDB 연동, Numpy/Pandas 기반 분석 가능

요구사항 고유번호		ECR-002
요구사항 명칭		DB관리 소프트웨어
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	스트리밍 및 재생 데이터를 저장, 조회하는 시계열 데이터베이스 관리 소프트웨어
	세부 내용	- 장비 품목 : DBMS(InfluxDB) - 장비 수량 : 1식 - 장비 기능 : 재생 데이터 수집 및 저장, 시나리오 결과 저장 - 장비 성능 및 특징 : Open API 지원, 시계열 DB 특화 구조, 태그 관리, Flux 언어 기반 데이터 가공

요구사항 고유번호		ECR-003
요구사항 명칭		시각화 대시보드 소프트웨어
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	작업자가 실시간으로 품질 및 생산 효율을 확인할 수 있는 대시보드 소프트웨어
	세부 내용	- 장비 품목 : Grafana (ECharts 플러그인 포함) - 장비 수량 : 1식 - 장비 기능 : 실시간 품질 점수 시각화, 목표 생산 달성률, 스팀 예측선/계산선 비교, 유사 공정 운전 궤적 표시, 변수 기여도 분석 등 - 장비 성능 및 특징 : Web 기반 시각화, 다양한 패널 구성 가능, 사용자 친화적 UI, 로컬 환경에서 즉시 실행 가능

요구사항 고유번호		ECR-004
요구사항 명칭		디지털 트윈 시뮬레이션 소프트웨어
요구사항 분류		시스템 장비 구성 요구사항
요구사항 상세 설명	정의	공정 데이터를 가상 3D 환경에서 시뮬레이션 및 시각화할 수 있는 소프트웨어 구성 요소
	세부 내용	- 장비 품목 : Unreal Engine (VaRest 플러그인 포함) - 장비 수량 : 1식 - 장비 기능 : 사용자 입력 기반 처리, API 연동을 통한 예측 결과 수신, UMG UI 기반 결과값 출력, 3D 기반 공정 시뮬레이션 제공 - 장비 성능 및 특징 : 실시간 API 요청/응답 처리, 로컬 환경에서 즉시 실행 가능, influxDB 데이터 연계 지원

3. 기능 요구사항

요구사항 고유번호		SFR-001
요구사항 명칭		데이터 스트리밍 및 저장 기능
요구사항 분류		기능
요구사항 상세 설명	정의	제지 공정 데이터를 초 단위로 재생 및 스트리밍하여 시계열 데이터베이스에 저장하는 기능
	세부 내용	- 장비 품목: Python 기반 스트리밍 모듈 - 1분 단위 데이터를 지정된 간격으로 재생 기능 - InfluxDB에 measurement 단위로 저장 기능 - 태그 기반 데이터 분류 기능 - LOT 단위 데이터 적재 및 재생 가능

요구사항 고유번호		SFR-002
요구사항 명칭		품질, 생산, 에너지 효율 분석 기능
요구사항 분류		기능
요구사항 상세 설명	정의	공정 데이터에서 산출된 품질, 생산성, 에너지 효율을 종합하여 단일 지표로 평가하는 기능
	세부 내용	- 기준값과 허용 범위를 활용하여 센서 제어 범위별 변동성 계산 기능 - 품질 관련 변수: 평량, 수분 - 생산 관련 변수: 속도, 지료 유량 - 에너지 관련 변수: 후건조기 압력 - 변수별 상관계수를 이용한 가중치 부여 기능 - 공정 단위를 초반/중반/후반 구간으로 나누어 구간별 중요도 반영 기능 - 최종 목적함수 y = Σ (가중치 × 변수값 × 구간별 가중치)로 산출 - $y = \sum_{j=1}^{3} a_j \cdot \frac{1}{\lvert T_j \rvert} \sum_{t \in T_j} \left(\sum_{i=1}^{n} w_i \cdot x_i(t) \right)$

요구사항 고유번호		SFR-003
요구사항 명칭		작업자 대시보드 시각화 기능
요구사항 분류		기능
요구사항 상세 설명	정의	분석 결과를 시각적으로 표현하여 작업자가 실시간 품질 상태와 개선 방향을 확인할 수 있도록 하는 기능
	세부 내용	- Grafana 기반 작업자 대시보드 제공 - 품질 점수 및 목표 생산 달성률 표시 - 스팀 사용량 예측값과 계산값 비교 그래프 제공 - 유사 공정 운전 궤적 비교 라인 차트 표시 - 변수별 기여도 파이차트, 배합 근접도 레이더차트 제공 - 허용 범위 초과 시 표시 및 권장값 제시

요구사항 고유번호		SFR-004
요구사항 명칭		디지털 트윈 시뮬레이션 응답 시간
요구사항 분류		기능
요구사항 상세 설명	정의	디지털 트윈에서 정보 입력 후 모델을 호출했을 때 결과가 출력되는 데 필요한 평균 응답 시간
	세부 내용	- Flask API 요청 시 응답 완료까지 평균 처리 시간은 1초 이내여야 함 - Unreal Engine UMG 화면에 결과 값 출력은 3초 이내에 완료되어야 함 - 전체 시뮬레이션 요청(입력, API 응답, 출력)은 5초 이내에 완료되어야 함

4. 성능 요구사항

요구사항 고유번호		PER-001
요구사항 명칭		데이터 스트리밍 처리 속도
요구사항 분류		성능
요구사항 상세 설명	정의	공정 데이터를 초 단위로 재생할 때, 데이터 누락 없이 시계열 DB에 저장하는 데 필요한 처리 속도
	세부 내용	- Python 스트리밍 모듈은 1분 단위 데이터를 5초 간격으로 변환 및 전송 - InfluxDB는 초당 수십 행 이상의 데이터 적재가 가능해야 하며 누락이 없어야 함 - Grafana 대시보드에서 실시간 패널이 5초 이내 갱신되도록 처리

요구사항 고유번호		PER-002
요구사항 명칭		품질, 효율 점수 산출 시간
요구사항 분류		성능
요구사항 상세 설명	정의	공정 운전 단위 데이터에 대해 다목적 함수 기반 품질, 효율 점수를 계산하는 데 필요한 평균 시간
	세부 내용	- 기준값 및 허용 범위를 활용한 변동성 계산은 1초 이내 처리 - 변수별 가중치 및 구간 가중치 반영 목적함수 계산은 공정 운전별 데이터 입력 후 5초 이내 완료 - 공정 운전 단위 결과 요약표 및 점수 시각화가 실시간으로 반영되어야 함

요구사항 고유번호		PER-003
요구사항 명칭		시각화 대시보드 응답 시간
요구사항 분류		성능
요구사항 상세 설명	정의	작업자 대시보드에서 조회 요청 시 결과가 표시되는 데 걸리는 평균 시간
	세부 내용	- Grafana 패널에서 데이터 쿼리 실행 시, 평균 응답 시간은 5초 이내여야 함 - 유사 공정 운전 궤적 비교, 파이차트, 레이더차트 등 복수 패널 구성 시에도 전체 화면 로딩은 5초 이내에 완료 - 데이터가 갱신되지 않거나 시각화 지연이 5초 이상 발생해서는 안 됨

요구사항 고유번호		PER-004
요구사항 명칭		디지털 트윈 시뮬레이션 응답 시간
요구사항 분류		성능
요구사항 상세 설명	정의	디지털 트윈에서 정보 입력 후 모델을 호출했을 때 결과가 출력되는 데 필요한 평균 응답 시간
	세부 내용	- Flask API 요청 시 응답 완료까지 평균 처리 시간은 1초 이내여야 함 - Unreal Engine UMG 화면에 결과 값 출력은 3초 이내에 완료되어야 함 - 전체 시뮬레이션 요청(입력, API 응답, 출력)은 5초 이내에 완료되어야 함

5. 인터페이스 요구사항

요구사항 고유번호		SIR-001
요구사항 명칭		품질, 효율 모니터링 대시보드
요구사항 분류		인터페이스
요구사항 상세 설명	정의	작업자가 품질 점수, 목표 생산 달성률, 주요 변수 상태를 직관적으로 확인할 수 있는 대시보드 제공
	세부 내용	- 공정 운전 단위로 품질 점수와 생산 달성률을 시각적으로 표시 - 품질 관련 변수(평량, 수분), 생산 관련 변수(속도, 지료 유량), 에너지 변수(후건조기 압력)를 한 화면에 배치 - 시계열 그래프를 통해 현재값과 허용 범위(band) 내외 여부를 구분 가능 - 목표 달성률은 게이지(Progress bar) 형태로 표시
주석		작업자는 생산 중 품질 변동을 신속히 파악할 수 있어야 하며, 허용 범위 초과 시 시각적 경고 표시가 필요함

요구사항 고유번호		SIR-002
요구사항 명칭		유사 공정 비교 및 운전 궤적 시각화
요구사항 분류		인터페이스
요구사항 상세 설명	정의	현재 운전 공정의 운전 데이터를 과거 유사 운전 공과 비교하여 운전 안정성을 확인할 수 있도록 시각화하는 기능
	세부 내용	- 공정 운전별 주요 변수 궤적을 중첩 라인 그래프로 표시 - 현재 공정과 유사 공정의 궤적 차이를 색상으로 구분 - 유사도 기준(평량, 수분, 속도 등)을 함께 표시하여 운전 조건 비교 용이
주석		작업자는 과거 유사 사례를 참고하여 현재 운전 안정성을 평가할 수 있어야 하며, 이상 변동 발생 시 빠른 판단이 가능해야 함

요구사항 고유번호		SIR-003
요구사항 명칭		변수 기여도 및 배합 근접도 시각화
요구사항 분류		인터페이스
요구사항 상세 설명	정의	공정 변수가 품질 점수에 미치는 영향도를 시각화하고, 배합비 근접도를 비교하는 기능
	세부 내용	- 공정 변수 영향도를 파이차트로 표시 (예: 지료 유량 40%, 수분 30% 등) - 배합 근접도는 레이더 차트로 표시하여 기준 배합과 현재 배합 간 차이를 직관적으로 파악 가능 - 각 변수의 허용 범위 대비 편차를 수치화하여 함께 제시
주석		작업자는 어떤 변수가 품질에 가장 큰 영향을 주는지, 배합 조건이 기준과 얼마나 다른지를 직관적으로 확인할 수 있어야 함

요구사항 고유번호		SIR-004
요구사항 명칭		디지털 트윈 입력 및 결과 시각화
요구사항 분류		인터페이스
요구사항 상세 설명	정의	디지털 트윈 환경에서 사용자가 데이터를 입력하고, 예측 결과를 직관적으로 확인할 수 있도록 하는 기능
	세부 내용	- LOT 및 시간을 입력할 수 있는 입력창 제공(UMG 위젯 형태) - 입력 결과(예측 스팀량, 품질 점수 등)는 같은 화면에서 즉시 확인이 가능하도록 구성 - 작업자가 별도 학습 없이 사용이 가능하고, 인지하기 쉽게 단순하게 구성하며, 결과값은 단순 수치 형태로 표시
주석		- 간단한 입력, 출력 중심으로 설계하여 시뮬레이션 결과를 작업자가 빠르고 쉽게 확인할 수 있도록 함

6. 데이터 요구사항

요구사항 고유번호	DAR-001
요구사항 명칭	공정 데이터 수집 및 저장
요구사항 분류	데이터
요구사항 세부내용	- 실제 공정 데이터는 보안 문제로 직접 활용이 어려움에 따라 원본 데이터의 구조와 분포를 유지하면서 랜덤 변동을 소량 부여한 시뮬레이션 데이터를 생성하여 사용 - InfluxDB에 bucket measurement 단위로 저장

요구사항 고유번호	DAR-002
요구사항 명칭	공정 데이터 정합성 검증
요구사항 분류	데이터
요구사항 세부내용	- 실제 데이터에서 파생된 시뮬레이션 데이터라도, 랜덤 변동 범위가 기준 허용치를 벗어나지 않도록 제어 - 데이터 타입, 단위, 해상도(분 단위/초 단위 등)는 원본과 동일하게 맞춤 - 결측치는 Forward Fill/Backward Fill 방식으로 보간 처리

7. 테스트 요구사항

요구사항 고유번호	TER-001
요구사항 명칭	데이터 스트리밍 및 적재 테스트
요구사항 분류	테스트
요구사항 세부내용	- Python 스트리밍 모듈이 생성된 데이터를 InfluxDB에 누락 없이 적재하는지 확인 - SECONDS_PER_MIN 파라미터에 따른 데이터 재생 간격 검증 (1분 단위 데이터 → 5초 단위 변환) - InfluxDB measurement별 데이터 적재 여부 확인 - 태그(role, kind, lot 등) 정상 기록 여부 검증 - 누락/중복 데이터 발생 시 로그 확인 가능 여부 확인

요구사항 고유번호	TER-002
요구사항 명칭	품질, 효율 점수 계산 정확성 테스트
요구사항 분류	테스트
요구사항 세부내용	- 다목적 함수 기반 품질, 효율 점수가 정확히 계산되는지 검증 - 기준값 및 허용 범위 계산 검증 - 변수별 가중치 및 구간별 가중치 반영 여부 확인 - 공정 운전 단위 종합 점수 산출 결과가 예상 범위 내에 위치하는지 검증

요구사항 고유번호	TER-003
요구사항 명칭	작업자 대시보드 UI 반응 테스트
요구사항 분류	테스트
요구사항 세부내용	- Grafana 대시보드 패널이 실시간으로 데이터를 표시하는지 확인 - 실시간 품질 점수, 목표 생산 달성률 패널이 5초 이내 갱신되는지 확인 - 스팀 사용량 예측선 및 계산선 그래프 동기화 여부 검증 - 유사 LOT 궤적 비교 라인 차트 정상 표시 여부 확인 - 기여도 파이차트, 배합 근접도 레이더차트 갱신 반영 여부 검증

요구사항 고유번호	TER-004
요구사항 명칭	디지털 트윈 시뮬레이션 연동 테스트
요구사항 분류	테스트
요구사항 세부내용	- Flask API 호출 시 입력값에 따른 예측 결과가 정상적으로 반환되는지 검증 - Unreal Engine UMG 입력창을 통한 사용자 요청 → API 응답 → 결과 출력의 절차가 정상적으로 수행되는지 검증 - 예측 결과(JSON)에 포함된 스팀 사용량, 품질 점수 값이 손실 없이 표시되는지 확인 - API 요청·응답 및 결과 출력 전체 흐름이 평균 5초 이내에 완료되는지 성능 기준 확인

8. 보안 요구사항

요구사항 고유번호	SER-001
요구사항 명칭	보안지침 준수
요구사항 분류	보안
요구사항 세부내용	- 개발 및 운영 과정에서 개인정보, 민감 데이터가 노출되지 않도록 관리 - ETRI 및 국가정보원 보안 규정을 준수하며, 데이터 처리 시 보안 정책을 반드시 반영

요구사항 고유번호	SER-002
요구사항 명칭	응용 및 DB보안
요구사항 분류	보안
요구사항 세부내용	- InfluxDB에 저장되는 데이터는 토큰 기반 인증을 통해 접근 제어 - DB 접근 로그 자동 기록, 관리자 이외의 직접 접근 제한

9. 품질 요구사항

요구사항 고유번호		QUR-001
요구사항 명칭		시스템 신뢰성
요구사항 분류		품질
요구사항 상세 설명	정의	시스템이 지정된 조건에서 일정 시간 동안 고장 없이 연속적으로 작동하고, 고장 발생 시 신속하게 복구될 수 있어야 함
	세부 내용	- Python 스트리밍 모듈, InfluxDB, Grafana 대시보드가 동시에 동작하는 환경에서 1시간 이상 무중단 운용 가능해야 함 - Flask API와 Unreal Engine 연동 시 예측 요청·응답이 95% 이상 성공률을 유지해야 함 - 데이터 누락·중복 발생 시 로그 자동 기록 제공 - 장애 발생 시 평균 복구 시간(MTTR)은 30분 이내로 제한

요구사항 고유번호		QUR-002
요구사항 명칭		사용자 운용 및 학습
요구사항 분류		품질
요구사항 상세 설명	정의	사용자가 시스템 기능을 빠르게 이해하고 쉽게 운용할 수 있도록 직관적인 UI와 학습 가이드를 제공해야 함
	세부 내용	- 신규 사용자가 30분 이내 튜토리얼을 통해 기본 기능(실시간 품질 점수 확인, 스팀 예측선/계산선 비교, 유사 공정 운전 궤적 확인 등)을 습득 가능해야 함 - 화면은 표, 그래프, 색상 등 시각적 요소 중심으로 구성하고, 복잡한 기능은 단계적으로 안내 - Unreal Engine UMG UI 입력창을 통해 입력값을 쉽게 설정할 수 있어야 하며, 튜토리얼에 디지털 트윈 시뮬레이션 실행 절차(값 입력 → API 호출 → 결과 확인)를 포함해야 함

요구사항 고유번호		QUR-003
요구사항 명칭		유지관리성
요구사항 분류		품질
요구사항 상세 설명	정의	시스템에 대한 변경 요구가 발생했을 때 빠르게 반영하고 유지보수할 수 있어야 함
	세부 내용	- 버킷 measurement 추가, Grafana 패널 수정 등 변경 사항을 1일 이내 반영할 수 있어야 함 - Python 모듈의 파라미터(SECONDS_PER_MIN, 허용 범위 값 등)는 설정 파일로 관리하여 쉽게 수정 가능해야 함 - Flask API 엔드포인트 변경이나 예측 모델 교체 시, Unreal Engine 연동 구성을 최소 수정으로 유지할 수 있어야 함

10. 제약 사항

요구사항 고유번호	COR-001
요구사항 명칭	개발 및 구현 환경 제약
요구사항 분류	제약사항
요구사항 세부내용	- 분석 및 시뮬레이션 기능은 Python 기반 프레임워크(Pandas, NumPy 등)를 사용해야 함 - 데이터 저장소는 InfluxDB(시계열 DB)로 제한하며, 다른 DBMS로의 전환은 지원하지 않음 - 대시보드는 Grafana 환경에서만 구현 가능하며, 작업자 대시보드 제공 - Flask API 기반의 연동 구조를 사용해야 하며, 다른 API 프레임워크로의 전환은 지원하지 않음 - Unreal Engine 5.3 환경과 VaRest 플러그인에 종속되므로, 다른 엔진이나 플러그인을 활용한 대체 구현은 제한됨

요구사항 고유번호	COR-002
요구사항 명칭	데이터 사용 제약
요구사항 분류	제약사항
요구사항 세부내용	- 실제 공정 데이터는 보안 문제로 직접 사용 불가 원본 데이터 구조와 분포를 유지하되, 랜덤 변동을 소량 부여한 시뮬레이션 데이터를 생성해 사용 - 분석 및 시각화의 검증은 해당 시뮬레이션 데이터를 기반으로 수행

11. 프로젝트 관리 요구사항

요구사항 고유번호	PMR-001
요구사항 명칭	수행 조직
요구사항 분류	프로젝트 관리
요구사항 세부내용	- 프로젝트 수행을 위해 다음과 같은 팀을 구성하고 역할을 명확히 해야 함 - 대시보드 팀: InfluxDB 기반 데이터 수집·저장 구조 설계. Python 스트리밍 모듈 구현 및 공정 운전 데이터 재생, 다목적 함수 기반 품질, 효율 분석 로직 개발, 작업자 대시보드 설계 및 시각화 (Grafana, ECharts 활용) - 디지털 트윈 팀: Unreal 기반 공정 시뮬레이션 환경 개발, 3D 시각화 및 가상 운전 화면 구현

요구사항 고유번호	PMR-002
요구사항 명칭	일정 계획
요구사항 분류	프로젝트 관리
요구사항 세부내용	- 계획된 프로젝트의 일정 및 단계별 산출물을 명확히 정의해야 함 - 1단계 – InfluxDB 환경 세팅, Python 스트리밍 모듈 초기 구현, 시뮬레이션 데이터(랜덤 변동 반영) 생성 및 적재 테스트 - 2단계 – 품질, 효율 분석 로직 개발, 공정 운전별 품질 점수 산출 및 검증, Flask API 서버 구현 및 디지털 트윈 연동 구조 개발 - 3단계 – 작업자 대시보드 설계, Unreal Engine UMG UI 및 API 호출·응답 결과 출력 검증 - 4단계 – 전체 모듈 연동 테스트, 스트리밍 데이터와 대시보드 실시간 동작 검증