

# 캡스톤 디자인 II 최종결과 보고서

프로젝트 제목(국문): 디지털 트윈을 활용한 스마트 팩토리 에너지 효율화 모델링 및 플랫폼 개발

프로젝트 제목(영문): Development of Energy Efficiency Modeling and Platform for Smart Factories Using Digital Twin

프로젝트 팀(원): EcoNOVA 학번: 20222024 이름: 박선아

프로젝트 팀(원): EcoNOVA 학번: 20221991 이름: 서지윤

1. 중간보고서의 검토결과 심사위원의 '수정 및 개선 의견'과 그러한 검토의견을 반영하여 개선한 부분을 명시하시오.

- 없음.

2. 기능, 성능 및 품질 요구사항을 충족하기 위해 본 개발 프로젝트에서 적용한 주요 알고리즘, 설계방법 등을 기술하시오.

• 생산량/품질 지표 도출을 위한 다목적 함수 설계

제지 공정의 생산량과 품질을 동시에 평가하기 위한 지표로 다목적 함수를 설계한다. 설계에 필요한 분석 데이터로 2022년 1월 1일부터 11월 30일까지 수집된 1분 단위 시계열 데이터(약 2,000개 변수)가 사용된다. 본격적인 분석 전에 공정 단위 길이가 30~60분인 구간만을 선별하고, 결측 비율이 높은 변수는 제거한 뒤 시간 순서를 고려한 Forward/Backward Fill 방식으로 결측을 보완한다. 또한 Rolling Z-score 기반 이상치 제거를 통해 시계열의 연속성을 확보한다. 사용되는 공정 변수는 속도와 지로유량(생산량 관련), 평량과 수분(품질 관련), 후건조기 압력(에너지 관련)으로 운전 변수의 핵심 센서값을 활용한다.

생산량과 품질을 종합적으로 고려하기 위해 다음과 같은 다목적 함수를 설계한다. 해당 다목적 함수는 시간 흐름에 따른 공정의 세부적 특성과 생산 품질 간의 상관성을 반영하여 설계된 구조이다. 다음과 같은 세 가지 주요 논리를 바탕으로 구성된다.

$$y = \sum_{j=1}^3 a_j \cdot \frac{1}{|T_j|} \sum_{t \in T_j} \left( \sum_{i=1}^n w_i \cdot x_i(t) \right)$$

$x_i$	시간 t에서의 공정 변수 i 값
$w_i$	변수 i의 중요도 (가중치)
$T_j$	전체 공정 시간의 구간 j (예: 앞, 중간, 뒤 15%/70%/15%)
$a_i$	구간 j의 품질 민감도 또는 분산 계수
$y$	시나리오의 생산성과 품질 종합 지표

그림 1. 다목적 함수

- 공정 구간의 분할 및 구간 가중치( $a_j$ ) : 제지 공정은 초반, 중반, 후반 구간마다 운전 안정성과 품질 특성이 다르게 나타나므로, 공정 단위를 15%, 70%, 15%로 분할하고 구간별 품질 민감도를 반영한 가중치를 부여한다. 이를 통해 품질이 안정된 구간에서 높은 생산성을 보이는 시나리오가 더 높은 평가를 받도록 한다.

- 공정 변수별 가중치( $w_i$ ) : 다목적 함수에 포함되는 변수는 영향도가 서로 다르므로, 각 변수와 생산량 간 상관계수를 산출하고 절대값을 정규화하여 변수별 가중치로 사용한다. 이를 통해 실제로 생산성과 품질에 기여하는 변수 중심의 평가가 이루어지도록 한다.

$$\text{생산량} = \text{지폭} \times \text{평량} \times \text{속도} \times \text{시간}$$

그림 2. 생산량 계산 수식

- 공정 구간별 시간 정규화( $\frac{1}{|T_j|}$ ) : 공정 단위마다 시간 길이가 다르기 때문에 구간별 값을 시

점 수로 나누어 평균화하는 정규화를 적용한다. 이를 통해 공정의 길이가 서로 달라도 공정하게 비교가 가능하도록 한다. 또한 실제 운전에서 존재하는 변수별 물리적, 운전적 한계값을 고려하여 제약조건을 설정함으로써 비현실적인 최적화 결과가 나오지 않도록 한다.

$$x_{i,min} \leq x_i(t) \leq x_{i,max} \quad \forall i, t$$

그림 3. 다목적 함수 제약 조건

다목적 함수의 성능을 확인하기 위해 높은 지표 값을 기록한 실제 공정을 확인한다. 생산량은 기존 평균 448.78 g/min에서 461.01 g/min으로 12.23 g/min 증가하여 약 2.7%의 생산성 향상이 있으며, 수분은 4.66%에서 4.63%로 0.03% 감소하여 품질 안정성이 유지된다. 평량은 기존 평균 161.65 g에서 166.93 g으로 5.28 g 증가하여 상품성이 향상된다. 후건조기 압력은 1.65 kg/cm<sup>2</sup>에서 1.54 kg/cm<sup>2</sup>로 감소하여 에너지 효율 측면에서도 효율적인 운전 상태임을 확인 가능하다.

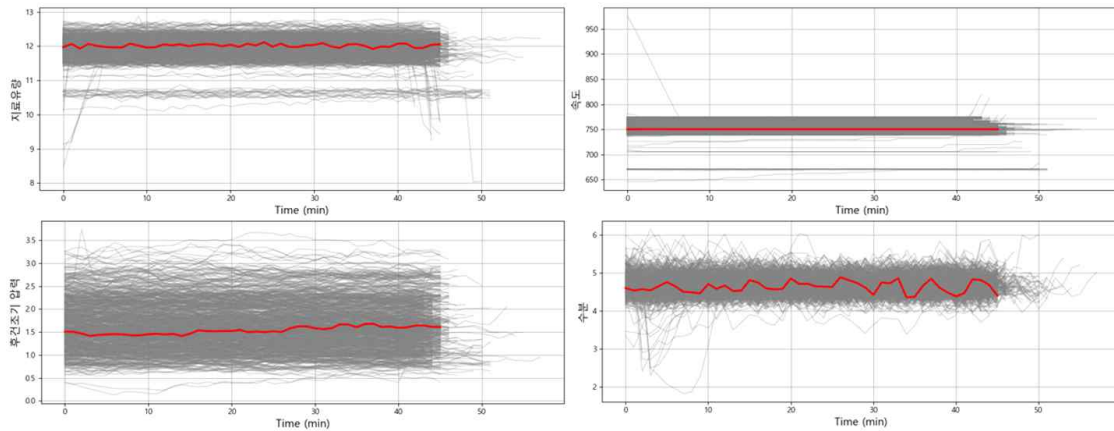


그림 4. 높은 지표값을 달성한 실제 공정 운전

또한 다목적 함수에 적용한 변수 조합의 타당성을 확인하기 위해, 높은 지표값을 보이는 공정에 대해 다항 회귀 곡면을 분석한 결과 해당 시나리오가 높은 목적함수 값을 갖는 영역에 위치함을 확인한다. 이는 제안한 지표가 실제 데이터의 분포 구조 안에서 일관성 있게 작동함을 의미한다.

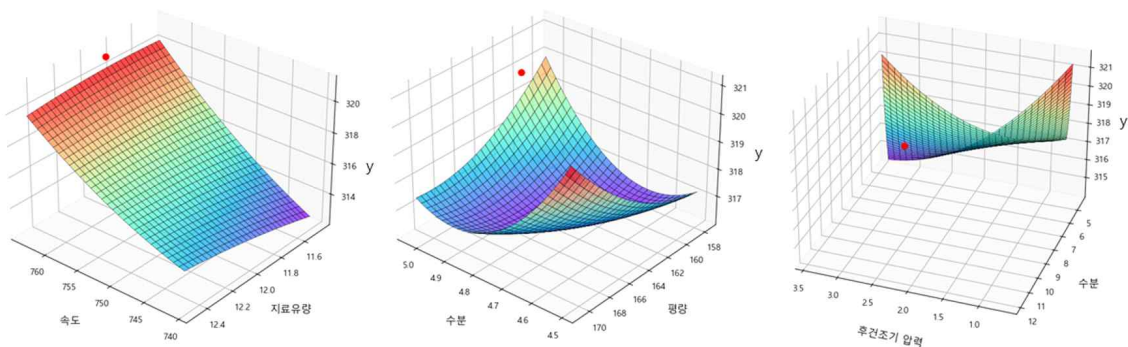


그림 5. 다목적 함수 기반 3D 다항 회귀 곡면

#### • 공정 센서 중요도/민감도 측정 알고리즘

제지 공정에 사용되는 주요 센서 변수가 공정에 미치는 영향을 정량적으로 분석하기 위해, 변수 중요도와 변수 민감도를 각각 계산한다. 중요도는 현재 공정 상태를 설명하는 데 각 변수가 얼마나 기여하는가를 평가하며, 민감도는 해당 변수를 조정할 경우 생산량/품질 점수가 얼마

나 변하는가를 측정한다.

변수 중요도는 현재 공정 데이터를 기반으로 계산된 다목적 함수 점수를 이용하여, 각 변수가 공정 성능에 어느 정도 기여하는지를 평가하는 방식으로 산출한다. 먼저 각 변수에 대해 기준값을 설정하는데, 기준값은 해당 변수가 정상 상태에서 나타내는 대표적인 중심값으로, 데이터의 평균이나 중앙값을 이용해 산정한다. 이후 모든 변수를 그대로 반영해 계산한 점수를 기준 점수로 사용하고, 특정 변수의 영향을 제거하기 위해 그 변수를 기준값으로 고정한 상태에서 다시 점수를 계산한다. 이때 기준 점수와 변수 고정 상태에서 계산된 점수의 차이는 해당 변수가 공정 성능에 미친 영향의 크기를 나타내며, 이러한 비교를 모든 변수에 대해 반복해 각 변수의 영향 정도를 구한다. 마지막으로 이 영향을 상대적 비율로 정리하여 최종 중요도로 표현함으로써, 특정 변수가 현재 공정 성능을 형성하는 데 어느 정도 기여했는지를 파악할 수 있다.

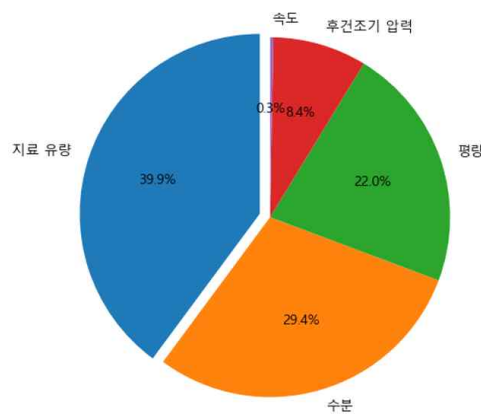


그림 6. 중요도 결과 시각화

변수 민감도는 각 변수를 실제 운전 범위 안에서 조금 변화시켰을 때 다목적 함수 점수가 어떻게 반응하는지를 분석하여, 공정 성능의 변화에 가장 민감하게 작용하는 변수를 파악하기 위한 지표이다. 이를 위해 먼저 기준 운전 상태에서의 점수를 계산하고, 이후 각 변수에 대해 허용범위의 10%에 해당하는 값을 변화량으로 사용하여 그 변수를 한 번은 증가시키고, 또 한 번은 동일한 폭만큼 감소시키는 두 가지 변형 데이터를 만든다. 이렇게 조정된 데이터로 다시 점수를 계산하면, 변수의 작은 변화가 공정 전체 점수에 어떤 방향과 크기로 영향을 주는지 확인할 수 있다. 증가와 감소 시점의 점수 차이를 변화량으로 나누어 얻은 값이 해당 변수의 민감도로, 이는 실제 공정에서 해당 변수를 미세하게 조정했을 때 성능이 얼마나 빠르게 반응하는지를 의미한다. 이러한 방식은 단순히 현재 상태의 기여 정도를 보는 것이 아니라, 제어 관점에서 어떤 변수가 조작 시 공정 전반에 더 강한 영향을 미치는지 판단할 수 있도록 도와준다.

- 펄프 조성 기반 공정 구조 분석 및 스팀 사용량 예측 모델 설계

제지 공정의 품질과 에너지 사용량과 펄프 배합비 변화의 관계를 파악하고, 이를 반영하기 위해 공정 구조 분석을 진행한다. 분석에 사용한 데이터는 2022년 1월 1일부터 11월 30일까지 수집된 1분 단위 시계열 데이터(약 2,000개 변수)로, 결측 비율이 높은 변수는 제거한다. 시간 순서를 고려하기 위해 결측치는 Forward Fill 방식으로 보완, 이상치는 Rolling IQR을 이용해 제거한다. 또한, 사용하는 펄프 배합비는 펄프 데이터로부터 각 구성 비율(%)을 직접 계산하여 산출하고, Silhouette Score 기반으로 최적 군집 수를 결정한 후 K-means 클러스터링을 진행하

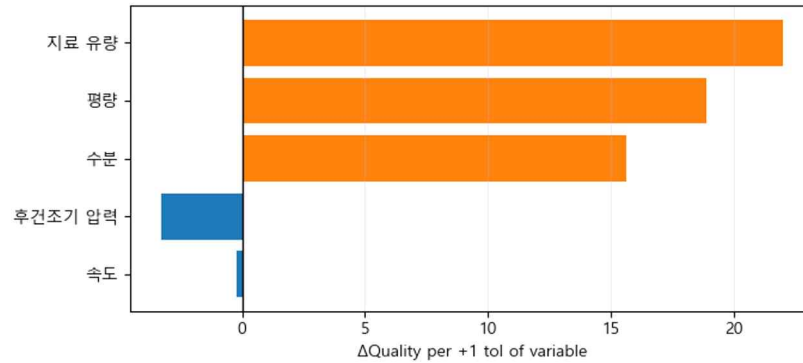


그림 7. 민감도 결과 시각화

여 유사한 조성 특성을 가진 공정 구간을 도출한다. 이후 특정 지종·평량 조건을 고정한 상태에서 클러스터링을 수행함으로써, 해당 제품군에서 안정적으로 운영되는 평균 펄프 배합비를 도출한다.

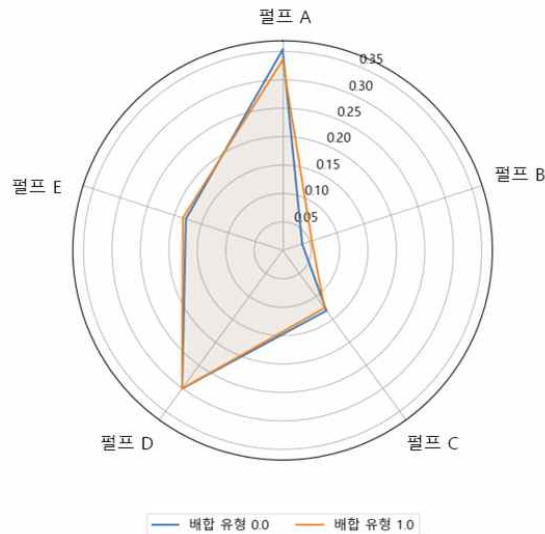


그림 8. 배합 유형별 평균 펄프 배합비 시각화

각 클러스터 별로 주요 변수의 중요도를 파악하기 위해 LightGBM 회귀 모델 기반 변수 중요도 분석을 적용하며, 이는 펄프 클러스터에 따라 공정 변수의 영향력이 달라지는 구조적 특성을 정량적으로 해석할 수 있도록 한다.

도출한 공정 구조와 변수 패턴은 LSTM 기반 스팀 사용량 예측 모델에 적용하여 공정 시계열 데이터(펄프 배합비, 속도, 수분, 평량, 압력 등)를 1분 간격으로 정렬하고 결측 보완·정규화 과정을 거쳐 LSTM 입력 구조를 구성한다. LSTM 모델을 학습하고, LOT 단위 운전 데이터를 기반으로 향후 스팀 사용량의 시계열 변화를 예측하며, 단순 센서 기반 예측보다 공정 구조를 반영한 정확도를 확보한다.

- 계절 변화 반영 대표 공정 및 통계 기반 제어범위 설계

제지 공정에서 계절별 온·습도 변화에 따른 건조 특성 및 에너지 소비 패턴을 파악하기 위해 계절별 대표 공정 분석을 진행한다. 클러스터링된 펄프 배합비를 기준으로 데이터를 계절별로 재구성하고, 클러스터 중심(평균 펄프 배합비)과의 Euclidean Distance와 z-score 기반 대표성 지표를 활용하여 각 계절에서 가장 안정적인 대표 공정 구간을 선정한다. 이때 상위 n% 중심성 기준과 최소 데이터 수 조건을 적용하여 대표 구간의 신뢰성과 통계적 유효성을 확보한다.

선정된 대표 공정 구간을 기반으로 주요 운전 변수(속도, 수분, 평량, 압력 등)에 대해 평균, 표준편차, 그리고 사분위 범위(IQR)를 산출하며, 이를 활용해 계절별 제어 권장 범위를 설계한다. 평균  $\pm 1\sigma$  범위는 정상 운전 변동성을 반영하고, IQR 범위는 안정 구간을 정의함으로써, 두 기준을 동시에 활용하는 이중 통계 기반 제어 범위 체계를 구축한다.

해당 계절별 대표 구간은 LSTM 예측 및 해석에 적용하여, 예측된 스팀 사용량이 정상 제어 범위 내에서 운전되고 있는지를 판단하는 근거로 사용한다. 이를 통해 공정 최적화 식의 제약 조건 뿐만 아니라, 실제 운전자가 계절 조건에 따라 조정해야 하는 데이터 기반 운전 가이드라인으로 활용할 수 있도록 한다.

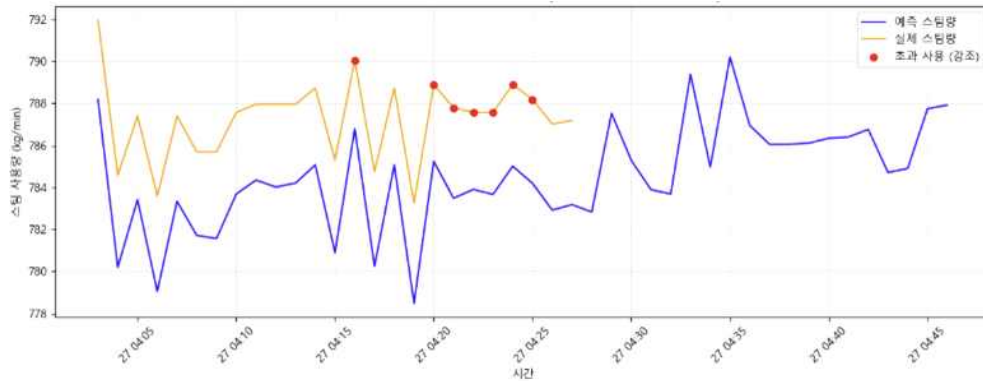


그림 9. 계절별 제어 권장 범위를 적용한 LSTM 스팀 사용량 예측 모델

3. 요구사항 정의서에 명시된 기능 및 품질 요구사항에 대하여 최종 완료된 결과를 기술하시오.

#### • 전체 시스템 구성도

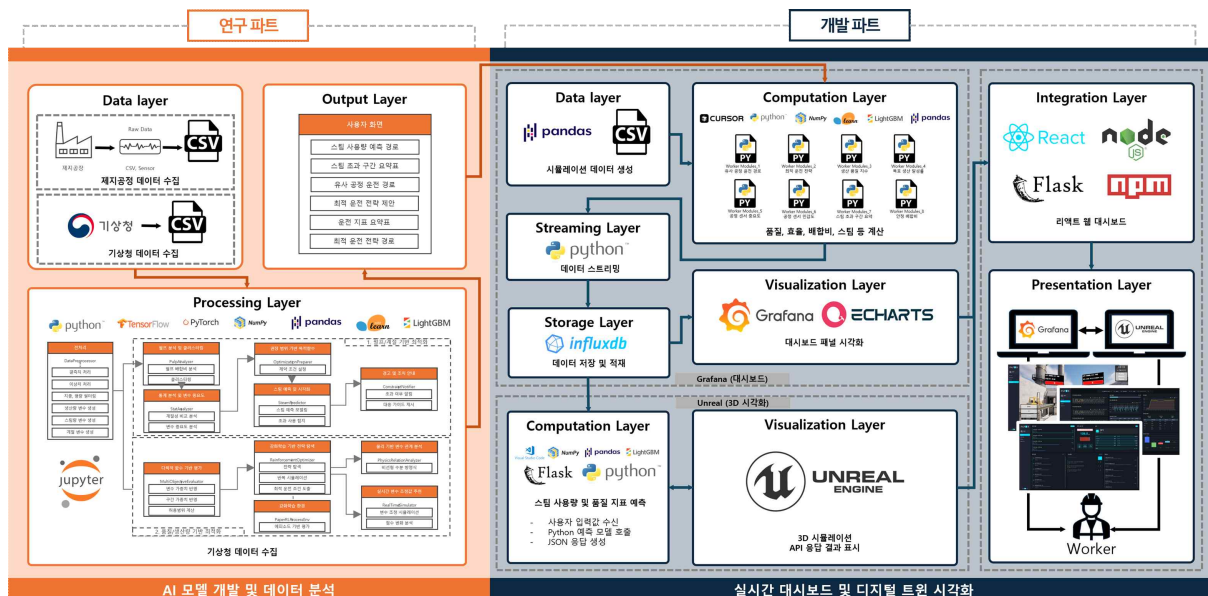


그림 10. 전체 시스템 구성도

• 웹 대시보드 시퀀스 다이어그램

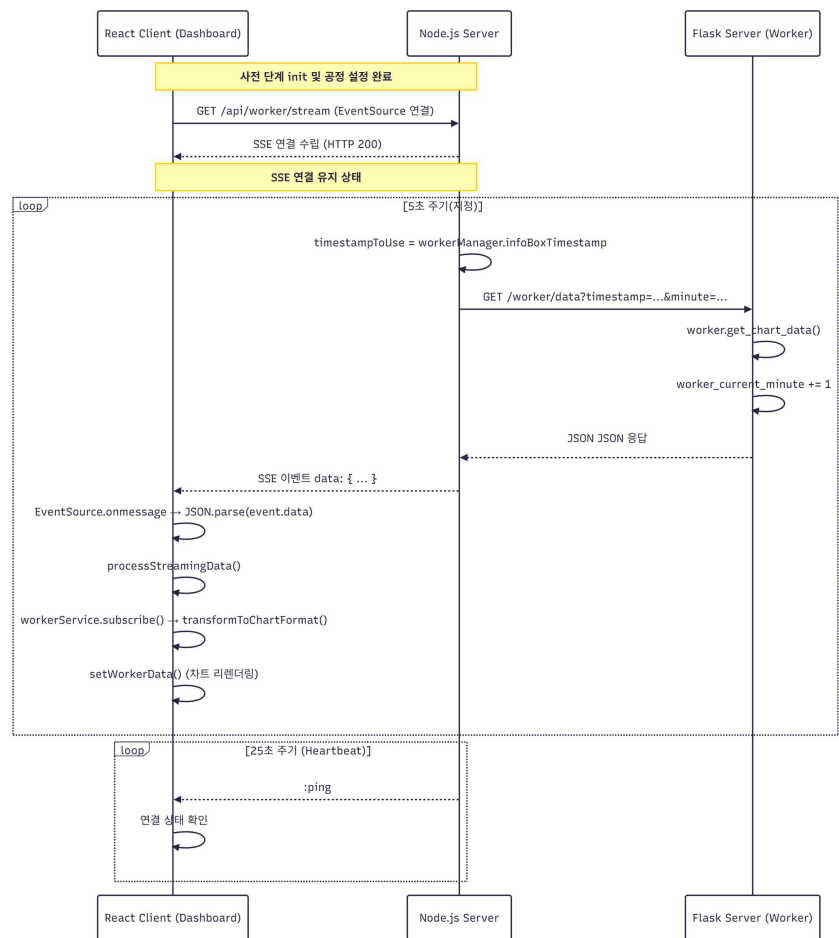


그림 11. 실시간 스트리밍 데이터 흐름 시퀀스 다이어그램

• 디지털 트윈 시뮬레이션 기능별 상세 요구사항

유스케이스ID	유스케이스명	사전조건	기본 흐름	예외	출력 결과
UC-01	시뮬레이션 초기화 및 HUD 표시	시뮬레이터 정상 로드	Unreal 실행 → HUD 생성 → 초기 위치 저장 → 기본 UI 표시	HUD 생성 실패 시 기본 화면 유지	HUD 패널 표시
UC-02	공정 시점 이동	HUD 정상 표시	이동 모드 확인 → 버튼 클릭 → 해당 위치로 이동	카메라/타겟 미존재 시 이동 취소	시점 전환 완료
UC-03	LOT/시간 입력 및 검증	HUD 입력란 활성화	LOT 입력 → 시간 입력 → 값 검증 후 저장	빈 값·오류 형식 입력 시 오류 메시지	유효한 입력값 저장
UC-04	Flask API 호출	LOT·시간 정상 입력	Unreal → Flask로 POST 요청 → Flask 응답 수신	서버 오류 또는 응답 없음	JSON 형태의 예측 결과
UC-05	Python 모델 실행	Flask 요청 수신	LOT 기반 데이터 로딩 → LSTM 예측, 품질 점수 계산 → JSON 생성	LOT 데이터 없음 → 오류 JSON	예측 및 계산 결과 반환
UC-06	예측 데이터 HUD, 3D UI 표시	Flask 응답 정상 수신	JSON 파싱 → HUD, 3D UI 스탬프 및 품질 점수 텍스트 갱신	JSON 파싱 실패 → 기존 HUD, 3D UI 유지	HUD, 3D UI에 예측 및 계산 결과 표시

표 1 디지털 트윈 시뮬레이션 기능별 상세 요구사항



- 디지털 트윈 시뮬레이션 모듈 구조도(모듈 명세서)

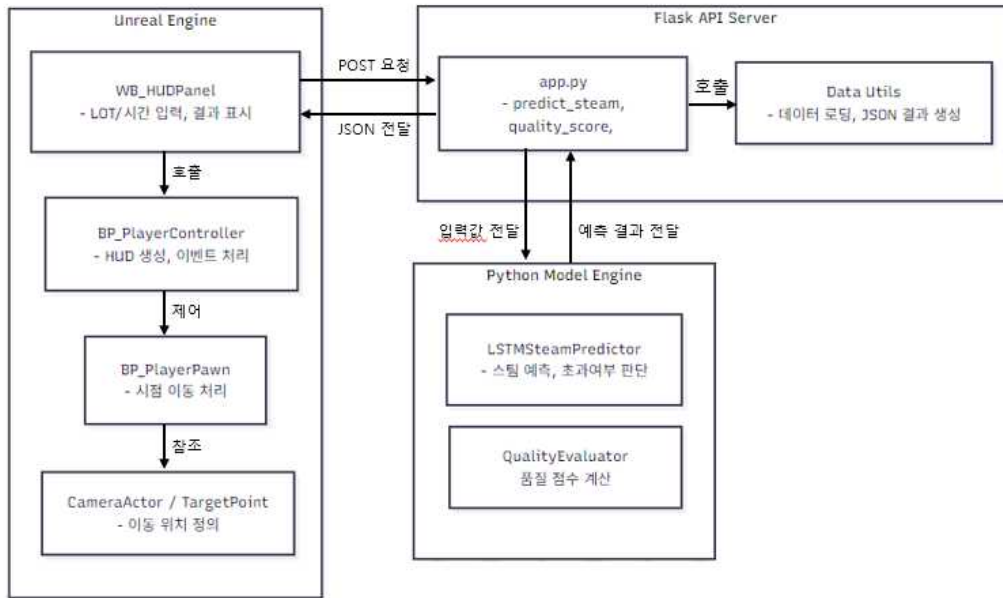


그림 12. 디지털 트윈 시뮬레이션 모듈 구조도(모듈 명세서)

- 구현하지 못한 기능 요구사항이 있다면 그 이유와 해결방안을 기술하시오,

최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
데이터 스트리밍 및 저장 기능	수정	대시보드 단순화와 보안 강화를 위해 InfluxDB 연동을 제거하고 Flask API 데이터만 사용하도록 변경
최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
품질, 생산, 에너지 효율 분석 기능	구현 완료	해당사항 없음
최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
작업자 대시보드 시각화 기능	수정	Grafana로 구현했으나 실제 사용자 활용 성과 배포 실효성이 떨어져 React 웹 대시보드로 전환
최초 요구사항	구현 여부(미구현, 수정, 삭제 등)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
디지털 트윈 시뮬레이션 응답 시간	구현 완료	해당사항 없음

- 요구사항을 충족시키지 못한 성능, 품질 요구사항이 있다면 그 이유와 해결방안을 기술하시오.

분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(성능) 데이터 스트리밍 처리	충족함(worker 모듈별)	해당사항 없음



속도	차이는 있으나 약 70~400ms 소요)	
분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(성능) 품질, 효율 점수 산출 시간	충족함( 공정 운전 수에 따라 차이는 있으나 1회에 15~65ms 소요)	해당사항 없음
분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(성능) 시각화 대시보드 응답 시간	충족함(초기 로딩 350~5,900ms, 렌더링 560~2,300ms, 업데이트 65~1,430ms 소요)	해당사항 없음
분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(성능) 디지털 트윈 시뮬레이션 응답 시간	충족함(API 응답, HUD 및 3D UI 표시에 약 65~80ms 소요)	해당사항 없음
분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(품질) 시스템 신뢰성	충족함(스트리밍과 서버 연동이 지속적으로 무중단으로 동작하며 응답 요청도 안정적으로 처리됨)	해당사항 없음
분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(품질) 사용자 운용 및 학습	충족함(사용자 가이드를 통해 신규 사용자가 기능을 이해하고 운용하는데 문제 없음)	해당사항 없음
분류(성능, 속도 등) 및 최초 요구사항	충족 여부(현재 측정결과 제시)	이유(일정부족, 프로젝트 관리 미비, 팀원 변동, 기술적 문제 등)
(품질) 유지관리성	충족함(독립적인 모듈 구조로 구성요소 간 결합도를 낮춰 수정·확장 시 영향 범위가 최소화되도록 설계됨)	해당사항 없음

6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

- Dashboard 실행 및 사용 방법

(1) 사전 준비

- Node.js (v18 이상)
- Python (3.8 이상)
- npm

(2) Node.js 의존성 설치

- cd Dashboard
- npm install

(3) Python 가상환경 설정 및 의존성 설치

- python -m venv venv
- .\venv\Scripts\Activate.ps1
- pip install -r requirements.txt

(4) 서버 실행

해당 명령어는 Express 서버, Python 워커 서버, React 개발 서버를 모두 동시에 실행합니다.

- npm run dev
- Python 워커 서버가 실행되지 않으면 Worker 관련 기능이 작동하지 않습니다.
- npm run dev를 사용하기 전에 concurrently 패키지가 설치되어 있어야 합니다 (npm install concurrently).

(5) 시스템 사용법

각 페이지에는 해당 기능을 이해할 수 있도록 사용자 가이드가 제공됩니다. 화면 좌측 하단의 사용자 가이드 버튼을 눌러 안내 내용을 확인하고, 제시된 설명을 따라 시스템을 쉽게 숙지할 수 있습니다.

• Digital Twin Simulation 실행 및 사용 방법

(1) 사전 준비

- influxDB (2.7.12 이상)
- Python (3.10 이상)
- Unreal Engine (5.3)

(2) InfluxDB 실행

- influxd.exe

(3) Python 가상환경 설정 및 패키지 설치

- python -m venv venv
- venv\Scripts\Activate.ps1
- pip install -r requirements.txt

#### (4) Flask 서버 실행

- python app.py
- Flask 및 Unreal 실행 전, InfluxDB 서버가 항상 켜져있어야 합니다.

#### (5) Unreal 플러그인 설정

Unreal 실행 후 다음 플러그인 활성화가 필요합니다(최초 1회만 필요).

- 메뉴 → Edit > Plugins → 검색 및 활성화
- ① VaRest(Flask API 연동용)
- ② HttpBlueprint(HTTP 요청)
- Unreal 플러그인 설정을 하지 않을 경우, 에셋 및 위젯 관련 블루프린트가 작동하지 않습니다.

#### (6) Unreal 실행 절차 요약

- ① Unreal Editor 실행
- ② 시뮬레이션 진행
- ③ 에셋 선택 → 좌측 하단 UI에 LOT, 시간 입력 → 전송시 결과값 UI 표시
- ④ 우측 상단 UI 선택 → 위치 이동 > 카메라모드 선택시 이동 불가, 카메라모드 미선택시 wasd 키로 이동 가능
- Unreal 실행 전 반드시 Flask API(python app.py)와 InfluxDB(influxd.exe)가 구동중이어야합니다.

### 7. 캡스톤디자인 결과의 활용방안

#### • 사회적 효과

- 작업자 지원 및 공정 안정화 : 직관적인 웹 대시보드와 운전 가이드를 통해 운전자의 숙련도에 관계없이 데이터를 기반으로 한 공정 운영을 가능하게 하며, 초보자도 안정적으로 설비를 운영할 수 있어 인적 오류를 줄이고 안전성과 대응력을 높일 수 있다.
- 탄소중립 및 ESG 대응 기반 마련 : 디지털 트윈 시뮬레이션을 통해 공정 조건에 따른 에너지 소비를 사전에 예측하고 대응하여 에너지 절감과 탄소 저감을 통한 친환경 공정 전환을 지원하며, ESG 경영과 기후변화 대응에 기여할 수 있다.

#### • 기술적 효과

- 디지털 트윈 기술의 산업 적용 확대 : 실제 데이터를 기반으로 한 디지털 트윈 시뮬레이션 구조는 이를 더욱 발전시켜 제조 현장의 교육, 운영, 설계 등에 폭넓게 활용될 수 있으며, 제조 산업 뿐 아니라 타 산업으로의 확장시킬 수 있다.
- 확장 가능한 플랫폼 구조 제공 : 모듈형 웹 대시보드와 시뮬레이션 시스템은 기능 추가 및 커스터마이징이 용이하여, 향후 AI를 기반으로 한 고도화나 자동제어 시스템 연동에 유리하다.

#### • 경제적 효과

- 에너지 절감 및 생산성 향상 : 다목적 함수와 강화학습 모델을 적용한 분석 결과 스팀 사용량은 최대 24% 절감, 분당 생산량은 약 2.8% 향상 가능성이 확인되었고, 이는 공장 비용의 절감에 기여할 수 있다.