

# 디지털 트윈을 활용한 스마트 팩토리 에너지 효율화 모델링 및 플랫폼 개발

Development of Energy Efficiency Modeling and Platform  
for Smart Factories Using Digital Twin

캡스톤 디자인 II 중간발표

팀명 : EcoNOVA  
팀원 : 박선아, 서지윤  
지도교수 : 이상금  
발표자: 서지윤  
발표일: 2025.09.10

#Digital Twin  
#Smart Factory  
#Optimization  
#Artificial Intelligence  
#Big Data  
#Web  
#Real-Time Monitoring



## CONTENTS

**01** 배경 및 필요성

**02** 목표

**03** 수행 방법 및 결과

**04** 향후 계획 및 개선 방향

**05** 구성도 및 역할

# Background & Necessity

## 배경 및 필요성

---



# 제지 산업의 특성과 디지털 전환 배경

## 에너지 다소비 산업

고온 스팀을 사용하는 에너지 다소비 산업 → 높은 에너지 소비와 탄소 배출.

## 환경 부담 및 대응 요구

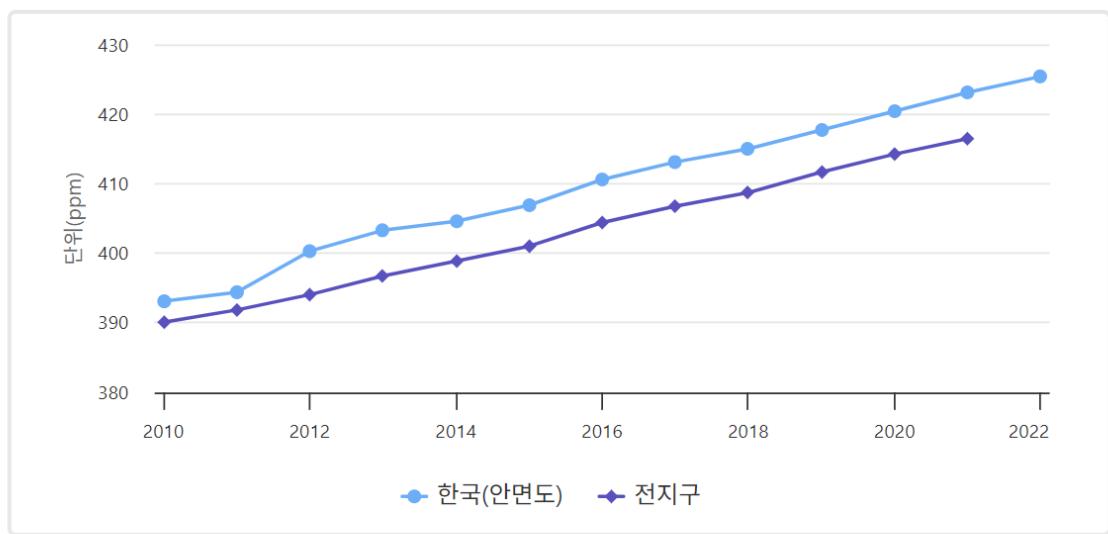
에너지 절감, 품질 개선 요구 및 친환경 · 저탄소 공정 전환이 필수 과제로 부상<sup>[1]</sup>.

## 디지털 전환 흐름

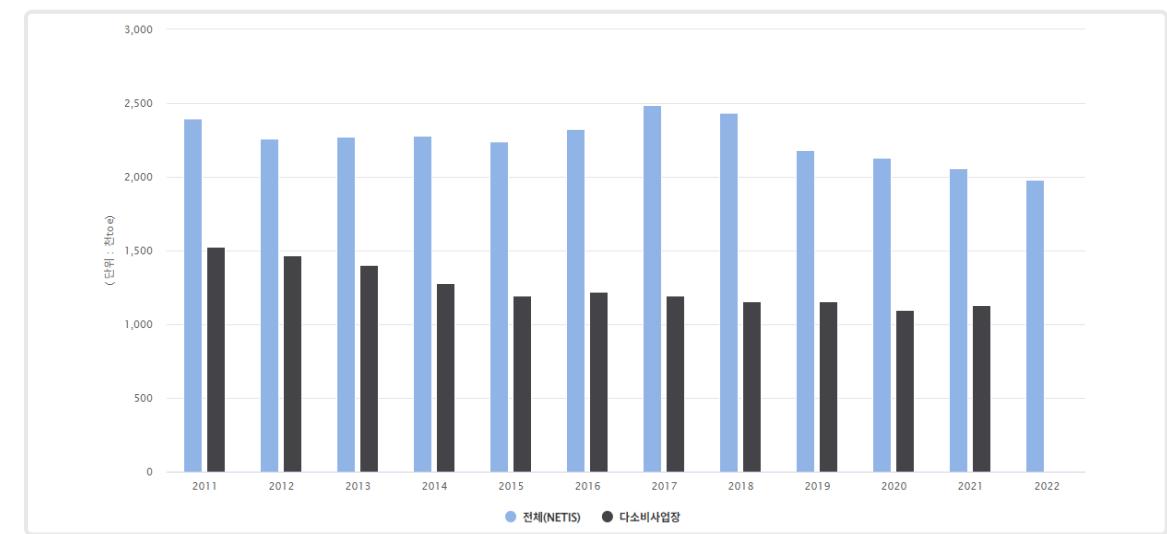
전 세계 산업계에서 디지털 트윈, 데이터 통합, AI 최적화 등 기술 도입 확대.

## 디지털 트윈 기술

공정을 가상 공간에 구현 및 시뮬레이션 → 에너지 · 품질 · 생산성 영향 사전 분석 가능<sup>[2][3]</sup>.



출처 : 탄소중립 정책포털



출처 : EG-TIPS 에너지 온실가스 종합정보 플랫폼

[1] A. Brown, P. Green, "Energy Efficiency in the Pulp and Paper Industry: Challenges and Opportunities," Energy Policy, vol. 149, 2021, (112345).

[2] Kamran Iranshahi, M. Gholami, S. Khosravi, "Digital twins: Recent advances and future directions in engineering fields," Lucerne University of Applied Sciences & Arts, Institute of Mechanical Engineering and Energy Technology, 2025, pp. 1-25.

[3] Experion Technologies, "Digital Twin for Smart Manufacturing," Experion Technologies White Paper, 2025.

# 공정 운영의 한계와 디지털 기반 최적화 필요성

## 최적화 연구 한계

일부 최적화 연구가 진행중이나 현장 적용 측면에서 실효성이 제한적.

## 데이터 관리 미흡

데이터 수집 및 통합 관리 체계 미흡<sup>[4]</sup> → 공정 데이터 기반 운영 어려움.

## 운전 방식의 한계

공정 데이터 분산, 경험 의존적 운전 방식 → 정량적 분석 및 의사결정 어려움.

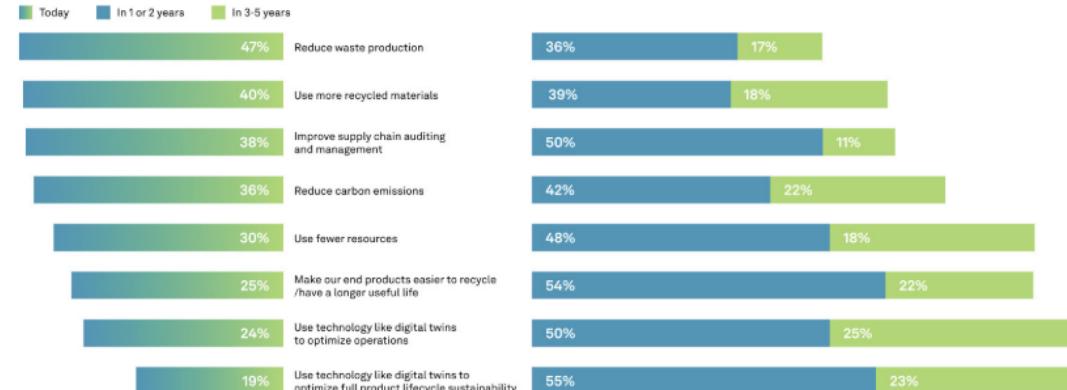
## 디지털 대응 필요성

공정 데이터 체계화, 디지털 트윈 기반 시뮬레이션 • 시각화 시스템 구축 필요<sup>[5]</sup>.



출처 : Siemens Amberg 공장

## When does your company plan to prioritise this initiative?



출처 : HEXAGON 2025 디지털 트윈 통계

[4] J. Smith, K. Lee, "Optimization of Pulp and Paper Manufacturing Processes: A Review," *Industrial & Engineering Chemistry Research*, vol. 59, no. 14, 2020, pp. 6543-6552.

[5] A. Florescu, "Digital Twin for Flexible Manufacturing Systems and Optimization Through Simulation: A Case Study," *Machines*, vol. 12, 2024, p. 785.

# Objectives 목표

---

2

# 목표

제지 공정 데이터를 통합 관리하고 분석하여 **에너지 절감과 품질 향상을 달성하는 것을 목표로 한다.** 이를 통해 비용 절감, 생산성 향상, 탄소 배출 저감 및 기후 변화 대응을 실현하며, 지속 가능한 제지 산업 발전을 위한 기술적 기반을 마련하고자 함.



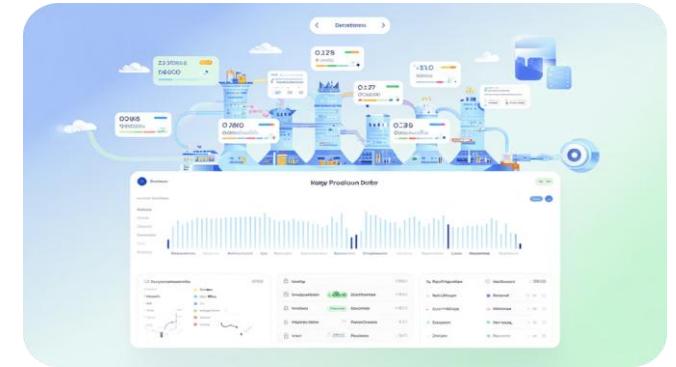
## 데이터 통합 및 DB 구축

- 펄프 및 계절 기반 최적화, 생산량과 품질 기반의 분석 모델을 바탕으로 공정 데이터를 통합 관리하는 DB를 구축.**
- 공정 이력, 품질, 에너지 사용 패턴 분석의 기반을 마련해 데이터 기반 의사 결정을 지원.**



## 가상 공정 시뮬레이션<sup>[6]</sup>

- 최적화 모델과 물리 기반 방정식을 활용해 가상 공정 환경 구현.**
- 시뮬레이션을 통해 에너지 소비와 품질 변화를 평가하고, 개선 효과를 분석하여 최적 운전 전략 도출.**



## 스마트 대시보드 개발

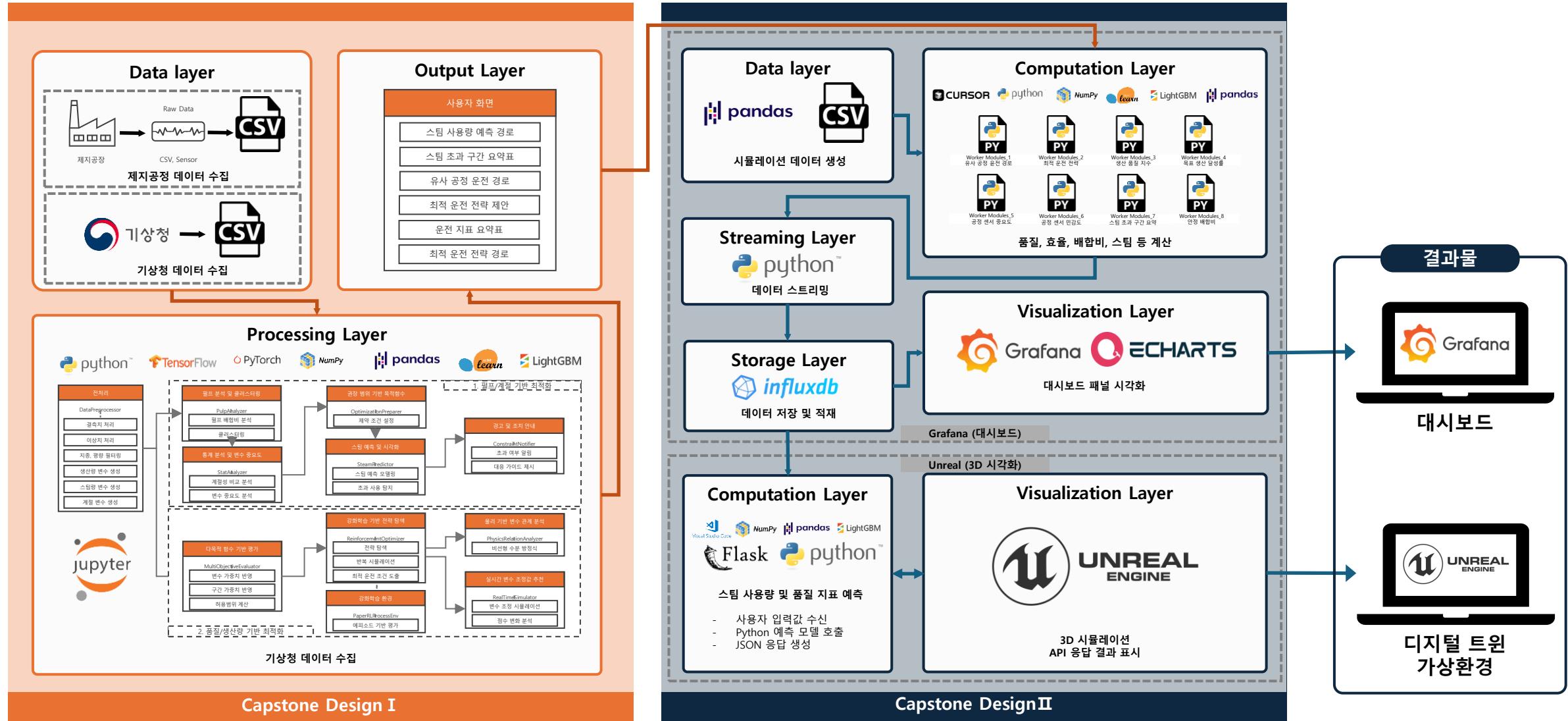
- 작업자에게 공정 상태와 운전 가이드 제공.**
- 작업자의 숙련도와 관계없이 누구나 안정적으로 공정을 운영할 수 있는 환경 조성.**

## Methods and Results 수행 방법 및 결과

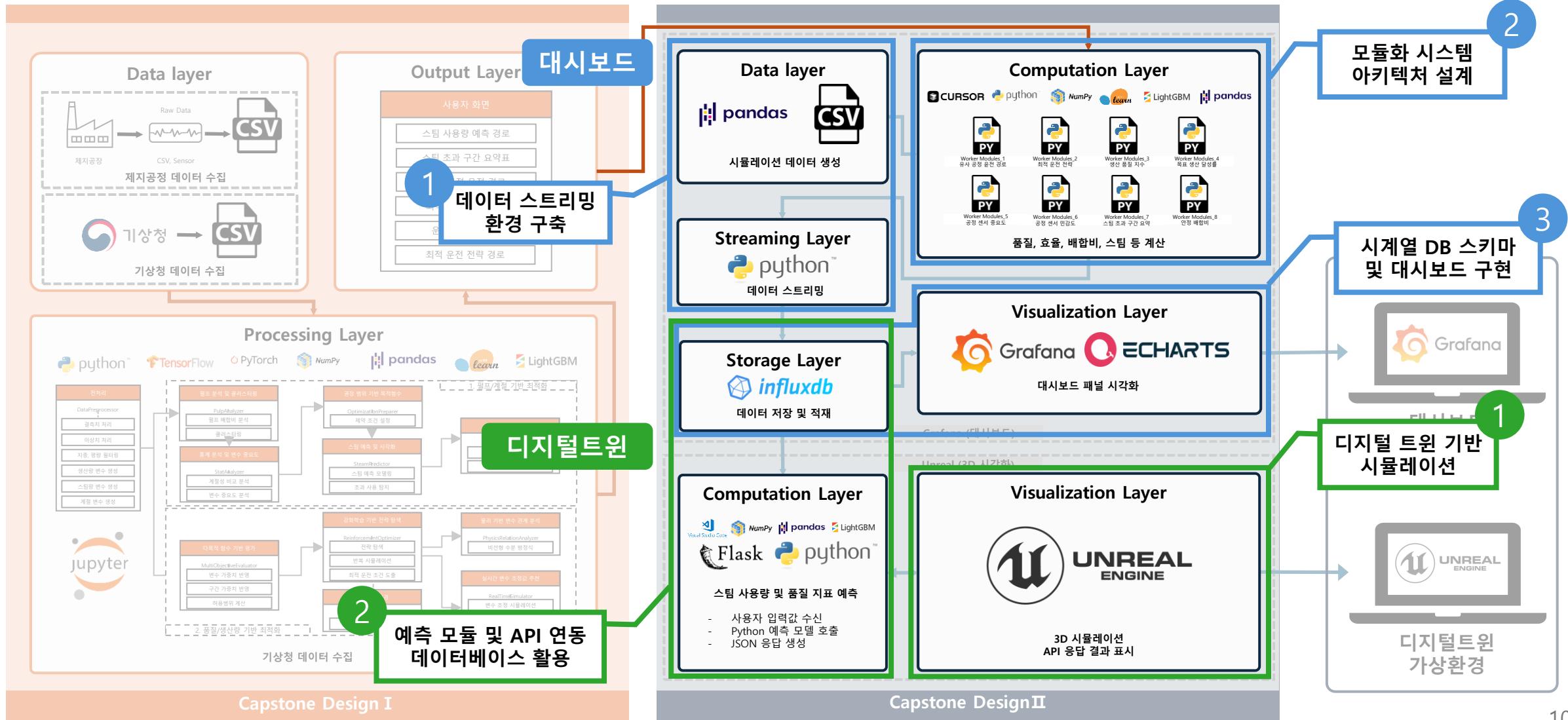
---

3

# 진행 상황



# 진행 상황



# 1 대시보드

## ● 1. 데이터 스트리밍 환경 구축

### 목적

: Python 모듈을 활용해 공정 운전 데이터를 1분 단위로 스트리밍하고 InfluxDB에 적재하여 Grafana에서 실시간 시각화 가능한 환경을 조성.

### 구성 포인트

- Python 스트리밍 모듈**
  - SECONDS\_PER\_MIN 파라미터로 1분 데이터를 가속/감속 재생.
  - 실행 로그로 데이터 적재 여부 확인 가능.

_field	_value	_start	lot	role
group	no group	group	group	group
	double	dateTime:RFC3339	string	string
score	79.6605744295608	2025-10-01T15:00:00.000Z	B3558	current
score	78.8094928665023	2025-10-01T15:00:00.000Z	B3558	current
score	81.21199764281253	2025-10-01T15:00:00.000Z	B3558	current
score	83.29255359051956	2025-10-01T15:00:00.000Z	B3558	current
score	83.8740308758146	2025-10-01T15:00:00.000Z	B3558	current

Python 스트리밍 모듈 실행  
(role=current 데이터 적재 확인)

```
18] ① 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)
```

공정 운전 데이터가 초 단위로 순차 전송되는 과정

### • 실행 환경 및 소프트웨어 구성

- 개발 환경: Python 3.10, influxdb-client, pandas, numpy  
(데이터 전처리 및 적재 안정성 확보)
- 데이터베이스: InfluxDB v2.7  
(시계열 데이터 최적화)
- Measurement-Tag-Field 구조 활용.
- 시각화 도구: Grafana v10.2 + ECharts 플러그인  
(레이더/파이/바 차트 지원)
- UTC 기반 타임스탬프 → Grafana에서 KST 변환 적용

### 문제 해결

#### • 문제1. 데이터 재생 불안정성

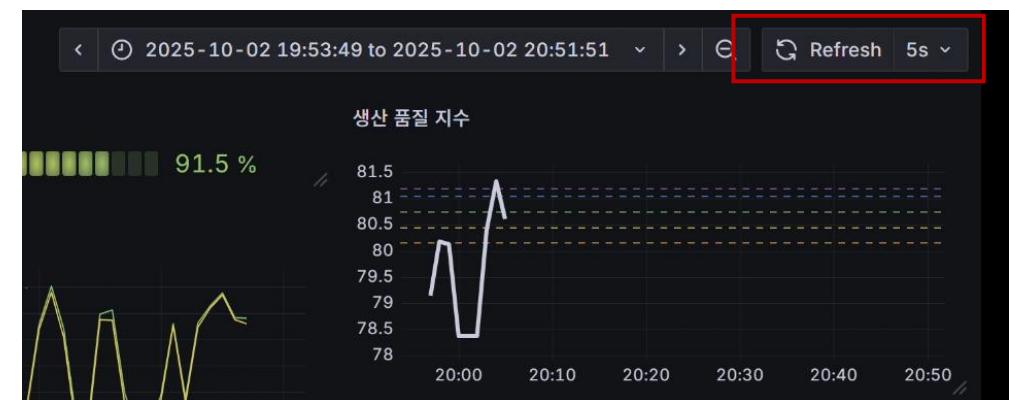
- 초기에는 CSV를 Grafana에 업로드해 정적 데이터만 확인 가능
- 실시간 모니터링 불가, 매번 수동 갱신 필요

⇒ Python 스트리밍 모듈을 제작하여 공정 운전 **데이터를 1분 단위 스트리밍 방식으로 전송**. SECONDS\_PER\_MIN 값을 통해 가속/감속 재생도 가능하게 하여, 테스트 환경에서 다양한 시나리오 검증 가능.

#### • 문제2. 실시간 반영 지연

- 초기 Grafana 패널은 즉각 반영 불가
- 수동 새로고침 필요

⇒ Grafana의 Refresh Interval 기능을 활용해 **5초 간격 자동 갱신을 설정**. Python 스트리밍 모듈과 연동 시, 데이터가 적재되는 즉시 대시보드가 업데이트되어 작업자가 실시간으로 품질 및 생산 상황을 파악할 수 있게 됨.



Grafana 대시보드에서 데이터가 5초 간격으로 자동 갱신되는 모습

## 2 대시보드

### ● 2. 모듈화 시스템 아키텍처 설계

#### 목적

: 공정 데이터를 단계별로 분석/처리할 수 있도록 모듈화된 시스템 아키텍처를 구축.

#### 구성 포인트

##### • 시스템 아키텍처

- 데이터 흐름: CSV 데이터 → Python Worker 모듈 → InfluxDB → Grafana
- Worker 모듈 역할 분담: 품질 지수, 효율 분석 등 기능별 분리
- 저장 구조 통일: 모든 결과를 Measurement 단위로 저장, 공통 태그부여
- 대시보드 연결: Worker별 Measurement를 읽어 패널 단위로 표시

##### • 모듈 설계 (Worker\_1 ~ Worker\_8)

모듈	주요 기능	입력	처리	출력 (InfluxDB)	목적(Grafana)
모니터링	유사 경로 운전 경로	CSV 시뮬레이션 데이터	유사한 운전 경로 탐색 및 전처리	유사 경로 데이터가 시간순으로 기록됨	유사 경로 비교
품질	최적 운전 전략	공정 운전 데이터	조건별 최적 전략 시뮬레이션	조건별 최적 전략 시뮬레이션	최적 전략 패널
생산량	생산 품질 지수	공정 운전 데이터	품질 지수 산출 (생蹲, 분산 등)	공정 운전 단위로 집계된 품질 점수	품질 지표 패널
생산량	목표 생산 달성을	생산 기록	목표 대비 달성을 계산	목표 대비 실제 생산 달성 정도	달성을 패널
공정 센서	공정 센서 중요도	센서 데이터	변수 중요도 분석 (기여도)	각 센서별 기여도 및 중요도 값	중요 변수 분석
공정 센서	공정 센서 민감도	센서 데이터	민감도 분석 (허용 범위 대비)	변수별 허용 범위 초과 정도와 민감도 결과	민감도 분석 패널
스팀량	스팀 초기 구간 요약	에너지 데이터	스팀 사용량 초기 구간 탐색	스팀 사용량이 초기와 구간 요약 정보	에너지 관리 패널
배합비	안정 배합비	공정운전 + 배합 데이터	안정적인 배합비 배합비 산출	안정적인 배합비 추천 결과	배합비 최적화 패널

Worker\_1~8 모듈 기능 및 데이터 흐름 요약

##### • 대시보드 패널 구조 설계

- Grafana 기반 작업자 대시보드 패널별 기능 분리.
  - 좌측: 유사 경로 모니터링 비교(다목적 접수 기반)
  - 상단: 목표 달성을, 스팀 예측 (핵심 생산성/에너지 지표)
  - 중앙: 최적 운전 전략 시뮬레이션, 품질 지수 (공정 관리 지표)
  - 하단: 센서 기여도/민감도, 배합비 (세부 분석 정보)

#### 문제 해결

##### • 문제1. InfluxDB 계산 한계 및 Python 모듈 보완

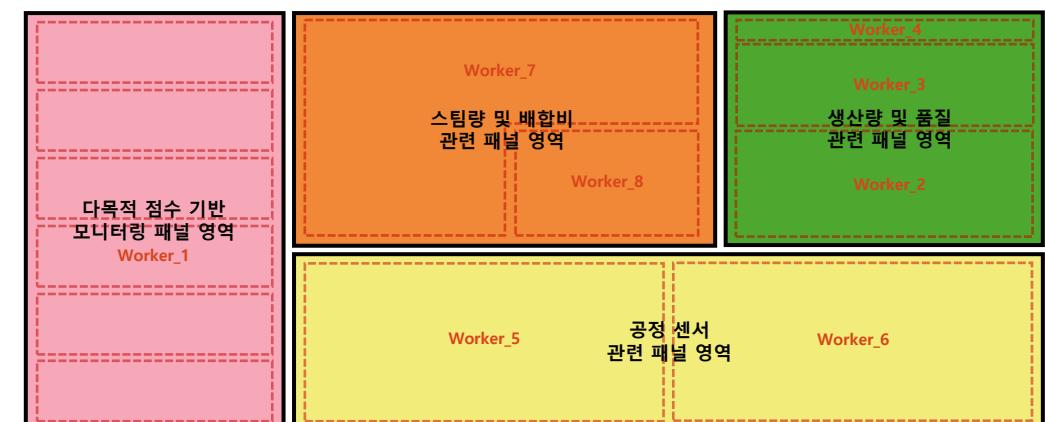
- InfluxDB는 시계열 저장소로 계산 기능이 제한적임  
⇒ 각 Worker 모듈에서 계산을 먼저 수행하고, 그 결과를 새로운 열로 추가하여 InfluxDB에 기록. 저장 시 공통 태그를 함께 부여하여 모듈 간 결과를 동일한 조건으로 조회 가능하게 함.

##### • 문제2. 대시보드 가독성

- 대시보드에 표시되는 패널이 많아 사용자가 원하는 정보를 빠르게 찾기 어려우며 가독성이 떨어짐

##### ⇒ 작업자 중심 설계

- 좌측(유사 운전) → 중앙(스팀/배합비) → 우측(생산/품질) → 하단(센서 분석) 순으로 계층적 배치를 적용.
- 작업자는 핵심 지표를 먼저 확인하고, 필요 시 세부 패널을 분석할 수 있어 효율적인 모니터링과 의사결정이 가능.



Grafana 작업자 대시보드 패널 레이아웃 설계도

### 3 대시보드

#### ● 3. 시계열 DB 스키마 및 대시보드 구현

##### 목적

: Worker 모듈별 산출 데이터를 통일된 DB 스키마에 저장하여 확장성과 유지보수성을 높이고, Grafana 시각화를 통해 복잡한 분석 지표도 직관적으로 표현.

##### 구성 포인트

###### • 데이터 저장 구조

- Worker별 Measurement(worker1\_m~worker8\_m)를 독립적으로 구성.
- Field에는 계산된 품질 점수, 효율, 스팀 사용량, 배합비 안정성 등 모듈 산출값을 기록.
- Tag는 최소한의 조건 필터링 요소(lot, paper, role, var 등)만 유지
- UTC 기반 저장에서 KST 변환으로 일관된 시간축 관리.

###### • Worker 모듈별 저장 스키마

WORKER1		WORKER2		WORKER3		WORKER4		WORKER5		WORKER6		WORKER7		WORKER8	
string	lot	string	lot	string	lot	string	lot	string	lot	string	lot	string	lot	string	lot
string	paper	string	paper	string	paper	string	paper	string	paper	string	paper	string	paper	string	paper
int	bw	int	bw	int	bw	int	bw	int	bw	int	bw	int	bw	int	bw
string	role	string	role	string	role	string	role	string	role	string	role	string	role	string	role
string	kind	string	kind	string	kind	string	kind	string	kind	string	kind	string	kind	string	kind
float	value	float	strategy_result	float	quality_index	float	production_rate	float	similarity_score	float	quality_score	float	steam_actual	float	stable_ratio
time	_time	time	_time	time	_time	time	_time	time	_time	time	_time	time	_time	time	_time

Worker 모듈별 저장 스키마 (InfluxDB 구조)

###### • 대시보드 패널 구조 구현

- 시계열 그래프: 품질 지수, 스팀 사용량 변화 추이
- 레이더 차트: 배합비 안정성 분석
- 파이 차트: 센서 중요도
- 바 차트: 민감도 분석 및 센서값 표기

Measurement	데이터의 큰 분류 (테이블 이름)
Tag	빠른 검색용 조건 (문자 정보)
Field	실제 값 (숫자/측정치)
Time	기록된 시간

\* Measurement-Tag-Field-Time 구조

##### 문제 해결

###### • 문제1. RDBMS와 시계열 데이터 부적합

- 기존 RDBMS(MySQL, PostgreSQL 등)를 고려했으나, 초 단위 데이터가 지속적으로 쌓이는 시계열 특성상 저장 및 조회 성능이 저하.

⇒ InfluxDB 채택. \*Measurement-Tag-Field-Time 구조를 활용하여 **대용량 시계열 데이터도 효율적으로 관리 가능**.

###### • 문제 2. 대시보드 패널 표현력 부족

- Grafana 기본 패널만으로는 배합비(다면량) 같은 레이더 차트 구현이 어려움.

⇒ ECharts 플러그인을 설치해 레이더 차트/파이 차트/바 차트 등 **다양한 형태**를 지원, 복잡한 데이터도 직관적으로 표현 가능.

##### 결과물



Grafana 대시보드의 확장 패널 (ECharts 적용 화면)

# 결과물: 대시보드



# 1 디지털 트윈

## ● 1. 디지털 트윈 기반 시뮬레이션

### 목적

- 제지 공정을 단순화한 가상 3D 환경을 구현하고, 공정 데이터 기반 입력값에 따른 결과 예측이 가능한 시뮬레이션 환경 구축

### 구성 포인트

- 3D 시뮬레이션 환경 구축**
  - Unreal을 활용해 주요 공정 구조를 단순화한 3D 시뮬레이션 구현
- 사용자 입력 UI 제공**
  - 사용자에게 입력, 실행, 결과 확인이 가능한 UI 제공
  - UMG 기반 UI 구성, IoT값 기반 주요 지표 상태 표시
- VaRest 플러그인 활용**
  - VaRest 플러그인을 활용해 Flask API 호출
  - 응답받은 JSON 데이터를 파싱해 UMG UI에 표시
- 데이터 흐름**
  - 사용자 입력 → Flask API 요청 → 예측 결과 수신 → Unreal UI에 시각화 데이터 출력

### 결과물



### 문제 해결

#### • 문제1. 디지털 트윈 구현 방식 변경

- 초기에는 Unreal에서 수식 기반으로 결과만 출력하는 시뮬레이션 구조 계획
- 기존 최적화 모델링의 단순 수식 표현, 전체 시스템과의 연계가 어려움

⇒ 예측 모델 및 결과 반영을 위해 Python 모델 + Flask API + DB 연동으로 구조로 전환. VaRest 플러그인을 통해 외부 예측 결과 수신 후 Unreal에서 UI에 표시. 모델 수정 시에도 유연한 적용 가능.

#### • 문제2. Unreal 블루프린트 연결 문제

- Unreal 사용 경험 부족으로 노드 실행선 및 타깃 연결 방식의 복잡함,
- 버전 변경에 따른 노드 이름·동작 차이로 Flask API 연동이 자주 실패함.

⇒ 전체 블루프린트 흐름도를 작성하고 주요 노드 연결 순서를 사전에 명확히 정의. 이를 통해 구현 중 오류 감소, 수정 시에도 고쳐야 하는 부분이 명확해져 안정적인 개발이 가능해짐.

## 2 디지털 트윈

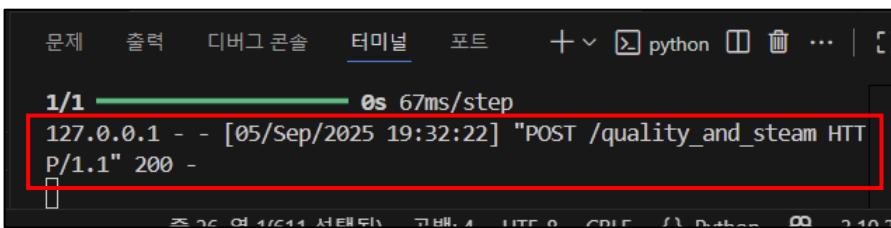
### ● 2. 예측 모듈 및 API 연동

#### 목적

: Python 기반 모듈을 Flask API로 구현하여 Unreal Engine과 연동된 분석 결과 제공

#### 구성 포인트

- **예측 모듈 구현**
  - 캡스톤1의 모델링 코드 기반 예측 모듈 구현
  - LSTM 기반 스팀 사용량 예측, 다목적 함수 기반 품질 점수 계산
- **API 엔드포인트 제공**
  - 기능별 결과를 요청-응답 구조로 제공
  - /predict\_steam : 스팀 사용량 예측
  - /quality\_score : 품질 점수 산출
  - /quality\_and\_steam : 통합 결과 제공
- **JSON 기반 응답 구조 제공**
  - 사용자 입력 요청에 따라 결과를 JSON 형식으로 반환
  - 예측 결과 항목 : lot, 시간, 예측 스팀량, 품질 점수 등
  - Unreal의 UI를 통해 예측 결과를 사용자에게 제공



Unreal → Flask API 데이터 요청 정상 처리 결과

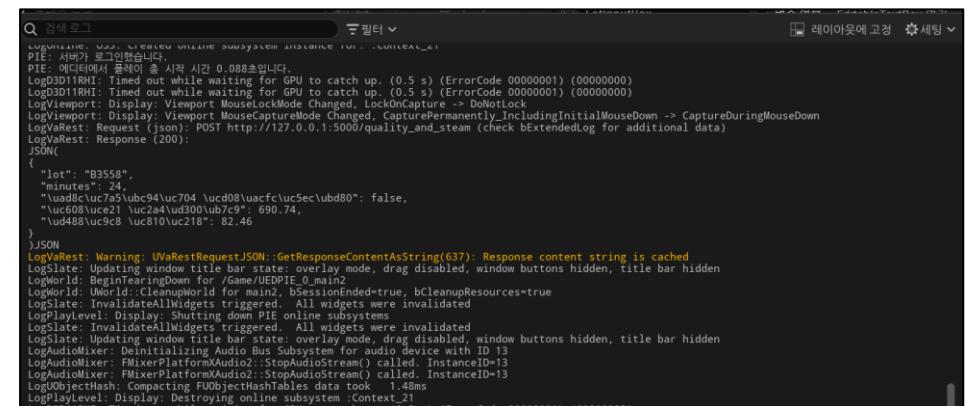
### ● 3. 데이터베이스 활용

#### 목적

: 예측 모델 학습 및 실행에 필요한 시계열 데이터 관리, LOT 단위 구조화 도움

#### 구성 포인트

- **시계열 데이터 구조화**
  - Measurement(paper\_data), Tag(LOT/지종/평량/계절), Field(주요 공정 변수, 계산 스팀량, 품질 관련 지표), Time(시계열 기준 시간) 형태로 저장
- **쿼리 및 변환**
  - Python 클라이언트를 통해 월 단위 쿼리 수행
  - 결과는 Pandas DataFrame 형태로 변환
- **예측 입력값 활용**
  - 변환된 데이터를 Python 예측 모듈과 Flask API가 참조
  - 학습 및 예측 시 필요한 입력값으로 활용



The screenshot shows the Unreal Engine's log window with the following text:

```
127.0.0.1 - - [05/Sep/2025 19:32:22] "POST /quality_and_steam HTTP/1.1" 200 -
{"lot": "B3558", "minutes": 24, "uad8cuc7a5\ubc94\ucd08\ucfc\ubc00\uc000": false, "u0c08\uec21\ue24\ud300\ub2e9": 690.74, "\ud488\uc9c8\ue80\uc218": 82.46}
)JSON
LogVaRest: Warning: UvaRestRequestJSON::GetResponseContentAsString(637): Response content string is cached
LogState: Updating window title bar state: overlay mode, drag disabled, window buttons hidden, title bar hidden
LogWindow: Dragging window title bar for main2, bSessionEnded=false
LogWindow: WindowId: Clearing world for main2, bSessionEnded=true, bCleanupResources=true
LogWindow: InvalidateAllWidgets triggered. All widgets were invalidated
LogPlayLevel: Display: Shutting down PIE online subsystems
LogPlayLevel: InvalidateAllWidgets triggered. All widgets were invalidated
LogState: Updating window title bar state: overlay mode, drag disabled, window buttons hidden, title bar hidden
LogAudioMixer: Destroying audio subsystem for instance with ID 13
LogAudioMixer: FMixerPlatform::Audio2::StopAudioStream() called. InstanceID=13
LogAudioMixer: FMixerPlatform::Audio2::StopAudioStream() called. InstanceID=13
LogObjectHash: Compacting F UObjectHashTable data took 1.49ms
LogPlayLevel: Display: Destroying online subsystem :Context_21
```

API 연동 테스트: Flask 응답값이 Unreal 블루프린트로 전달되는 과정 로그

## 결과물: 디지털 트윈 가상환경



## Future Plans & Improvements 향후 계획 및 개선 방향

---

4

# 시스템 연동

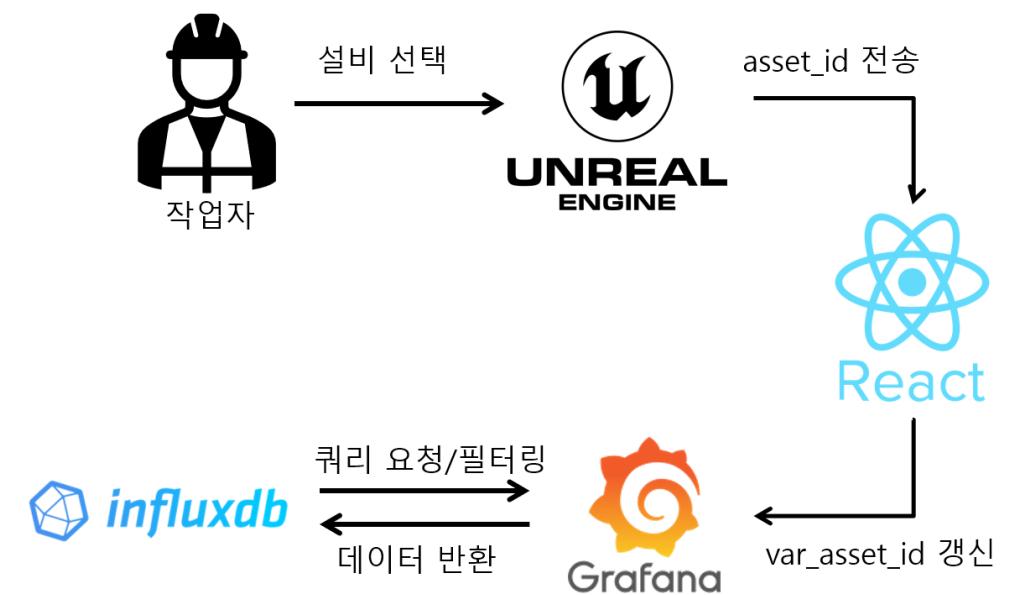
## ● Unreal –Grafana 대시보드 연동

### 목적

: 3D 환경(UE)에서 설비(asset)를 선택하면, Grafana 대시보드 내 관련 패널이 강조(테두리/하이라이트 표시) 되어 직관적으로 식별 가능

### 구현 방식

- 1. 설비(asset) 단위 시뮬레이션 구현
  - 기존 Unreal에서 단일 설비에서만 진행하던 시뮬레이션 구조 개선
  - 각 설비별로 기능(스팀•품질•제어 등)을 구분하여 3D 시뮬레이션 반영
- 2. 게이트웨이-React 연동 구조 도입
  - Unreal에서 특정 설비 선택 시 asset\_id 변수를 게이트웨이 서버로 전송
  - 서버는 React 컨테이너를 통해 Grafana 임베드 URL내 var\_asset\_id 변수 갱신
  - Grafana 대시보드가 필터링되어 해당 설비의 상세 지표 표시
- 3. 양방향 강조 효과 적용
  - Unreal : 선택된 설비에 Outline/Material 하이라이트 적용 → 시각적 구분
  - Grafana : 해당 설비 관련 패널 그룹화 및 강조 표시 → 직관적 식별성 강화
  - 두 환경(3D + 대시보드) 간 시각적 연결성 강화
- 4. 동적 쿼리 필터링 체계 활용
  - 기존 InfluxDB 태그 체계(lot, paper, bw 등)를 활용하여 동적 쿼리 필터링 구성



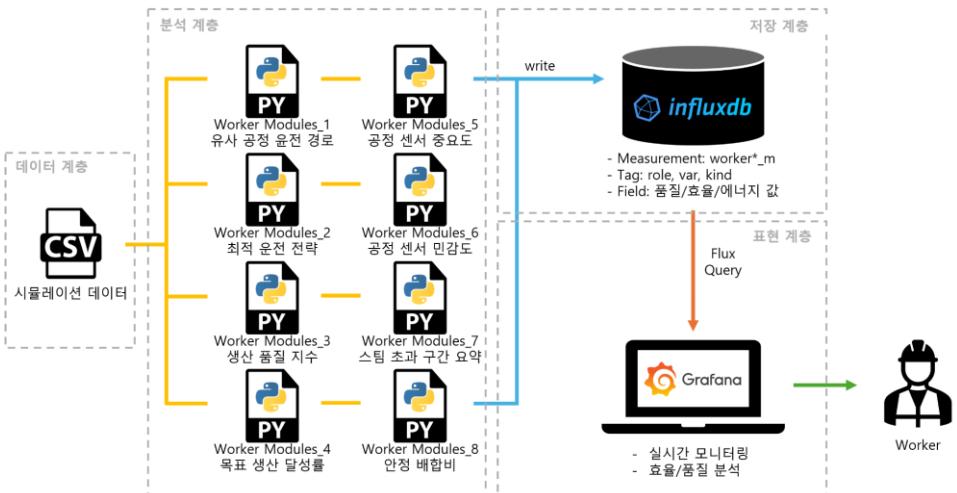
# System Architecture and Roles

## 구성도 및 역할

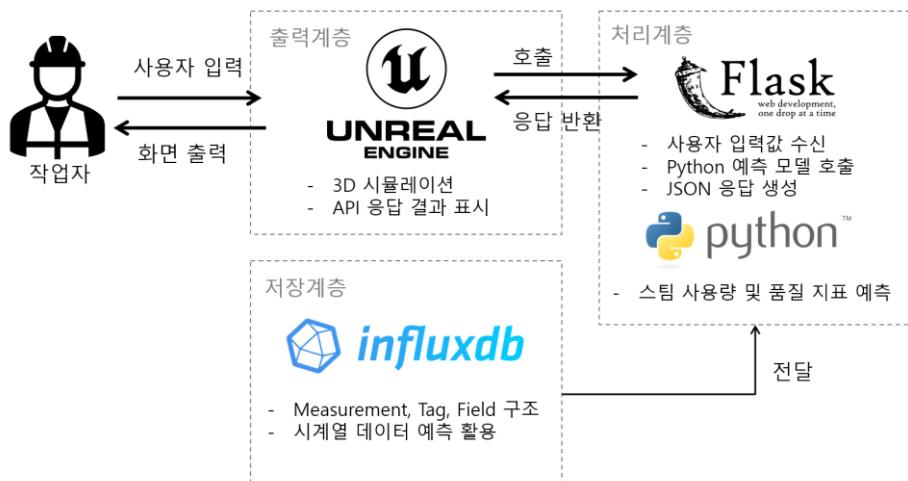


# 담당 역할

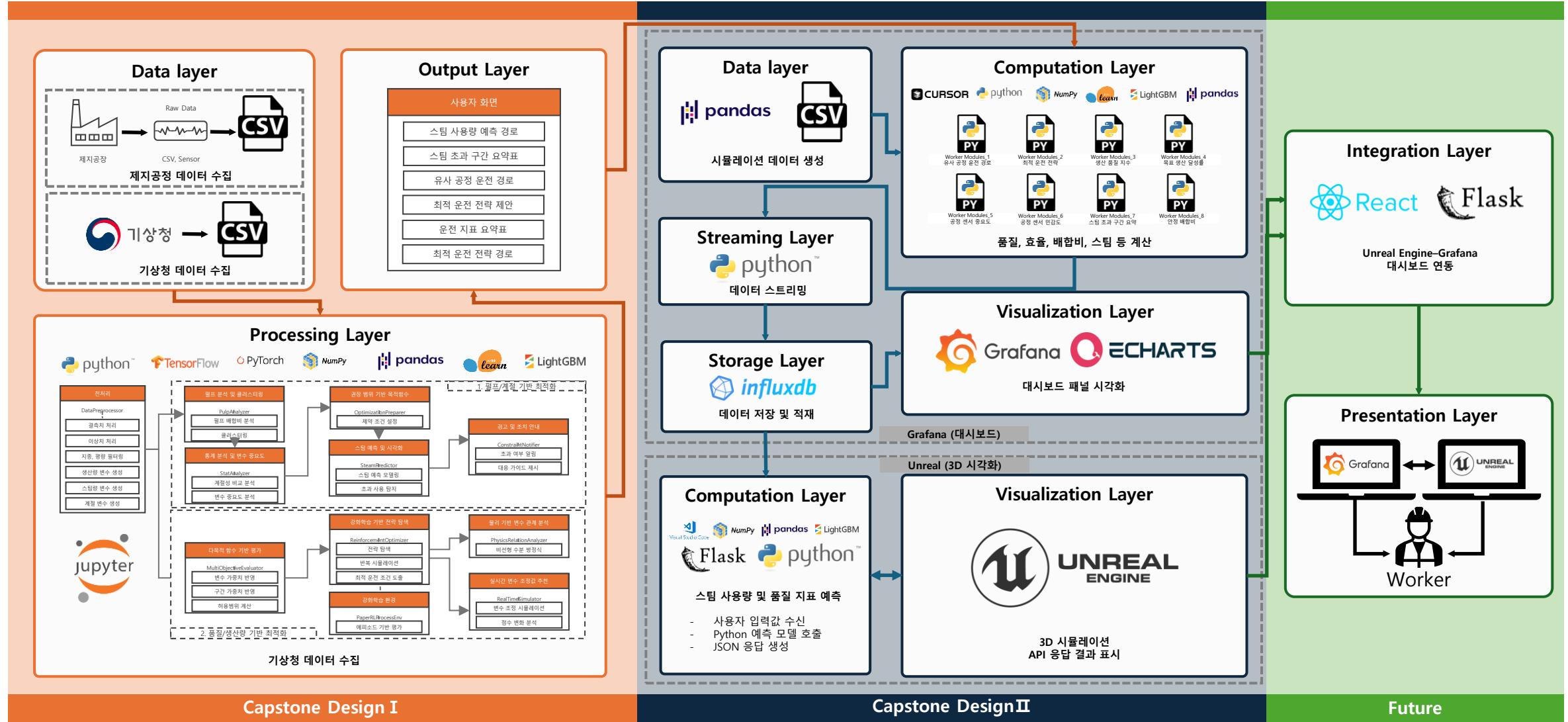
 <p><b>박선아 (팀장)</b></p>	전처리	품질, 생산량 기반 작업자 시각화 정리
		Python 기반 데이터 스트리밍 모듈 구현
	Worker	Worker 모듈 설계 및 기능 분리
	DB	InfluxDB 기반 시계열 데이터베이스 구축
	시각화	Grafana 기반 작업자 대시보드 설계 및 패널 구성



 <p><b>서지윤 (팀원)</b></p>	전처리	스팀 기반 작업자 시각화 정리
		모델링 기반 스팀 사용량, 품질 점수 모듈 구현
	API 연동	Flask API 구축 및 엔드포인트 설계
	DB	InfluxDB 기반 데이터 저장 구조 설계 및 연동
	시각화	Unreal Engine 기반 3D 시뮬레이션 환경 구축



# 전체 구성도



## 학술대회 참여



The image is a collage of four distinct scenes. The top left shows a steep, rocky cliff overlooking a body of water. The top right is a close-up of a small, green, circular island with a crater-like shape. The bottom left features a traditional Korean stone statue, possibly a jangseung (spirit guardian). The bottom right is a wide-angle night photograph of a modern city or resort area with numerous lights and a large swimming pool.

국립한밭대학교, 한국전자통신연구원

# 스마트 팩토리 운전 전략 최적화를 위한 다목적 함수 및 강화학습 기반 전략 도출

## Deriving Optimal Operational Strategies for Smart Factories Using Multi-Objective Functions and Reinforcement Learning

### 박선아<sup>1</sup>, 서지운<sup>2</sup>, 도윤미<sup>2</sup>, \*이상금<sup>1</sup>

### \*국립한밭대학교, 한국전자통신연구원

### Eco AI Lab

## Abstract

제작 공정은 시계별 변동성과 변수 간 복잡한 상호작용으로 인해 철학적인 운전 전략 수립이 요구된다. 본 연구에서는 생산성과 품질, 에너지 효율을 동시에 고려하기 위해 다목적 함수 기반 최적화 방식과 강화학습 기반 전략 학습 방식을 적용하고, 실제 공장 데이터를 활용하여 그 성능을 비교 분석. 실험 결과, 다목적 함수 기반 시나리오는 생산량을 2.8% 증가시키고 수분율을 0.64% 감소시키며, 후진기 압축률은 6.67% 낮추면서 전반적인 균형 형상을 달성. 강화학습 기반 시나리오는 생산량의 소폭 증가와 함께 수분율을 1.72% 감소시키고, 후진기 압축률을 24.24% 줄이며 품질 만족도와 에너지 효율을 측면에서 두 개의 시나리오를 보임. 두 방식은 각각 다른 강점을 가지며, 실제 산업 공장에서 운전 전략을 선택하고 설계하는 데 있어 유의미한 기준을 제공할 수 있음.

## 1. Introduction

### ESG 경영 확산

- 최근 제조 산업은 ESG 경영 확산에 따라 에너지 효율과 자원 최적화가 중요해져 스마트팩토리 운전 전략 효율화가 핵심 과제로 떠오르고 있음.

### 제작 공정 설정

- 제작 공정은 필름을 원료로 하는 고속 응집공정으로, 각 단계에서 다양한 센서 데이터가 실시간으로 수집됨.

### 제작공정 운전 및 최적화 필요성

- 공정 변수는 시계에 따라 복잡하게 작용하며, 품질과 생산성에 큰 영향을 미침.
  - ex) 속도와 양수 분분 등에서 변화가 충돌 발생과 에너지 낭비에 이어질 수 있음.
- 기존의 제작자 경험 기반 제작 방식은 복잡한 변수 상호작용을 방지하기 어려워, 따라서 본 연구에서는 다목적 함수 기반 최적화와 강화학습 기반 운전 전략 학습을 적용함.

### 제작공정 운전 및 최적화 필요성

- 공정 변수는 시계에 따라 복잡하게 작용하며, 품질과 생산성에 큰 영향을 미침.
  - ex) 속도와 양수 분분 등에서 변화가 충돌 발생과 에너지 낭비에 이어질 수 있음.
- 기존의 제작자 경험 기반 제작 방식은 복잡한 변수 상호작용을 방지하기 어려워, 따라서 본 연구에서는 다목적 함수 기반 최적화와 강화학습 기반 운전 전략 학습을 적용함.

## 2. Multi-Objective Function

### 구간별 변동성 발생

- 제작공정 데이터를 분석한 결과, 지층 및 평방이 변경되는 시점에서 변수들 및 급격한 변동성이 발생.

(a) 구간: 공정이 안정적인 상태 유지.  
(b) 구간: 공정이 변동성이 발생. 이후 변수 불안정화와 품질 변동성 급증.

- 따라서 운전 전략 설계 시 구간별 변동성 특성을 반영하는 것이 중요.

### 다목적 함수 정의

$$y = \sum_{i=1}^3 \beta_i \cdot \frac{1}{T_j} \sum_{t=1}^T \left( \sum_{k=1}^n w_k \cdot x_k(t) \right)$$

subject to  $x_{l,\min} \leq x_l(t) \leq x_{l,\max}$

변수	설명
$w_i$	변수 i의 중요도로 차지하는 가중치 (제작공정의 상대적인 기여도)
$T_j$	전체 운전 시간 기준으로 나누어 구간 (초단위/총시간 = 15% / 100%)
$\beta_i$	구간별로 중요도를 갖는 가중치

• 이 점수  $y$ 를 기준으로 다양한 운전 조건의 시나리오를 비교·평가함.  
• 제작조건을 통해 변수 값의 물리적 현실성을 보장함.

## 3. Reinforcement Learning

### 강화학습 사용

- 제작 공정은 변수 간 상호작용이 복잡하고 상태가 지속적으로 변함.
- 기존 수식 기반 최적화는 시계별 변화에 유연하게 대응하기 어려움.
- 따라서 전략을 스스로 학습하여 환경 변화에 적응할 수 있는 강화학습을 적용.

### 강화학습 설계 구조

Agent → Action → Environment → Reward → Model → Agent

- 에이전트(Agent): 공정 상태(State)를 받아 제어 행동(Action)을 선택하고 학습.
- 환경(Environment): 속도, 유량, 압력 등 주요 변수로 구성.
- 보상 함수(Reward): 공정을 소진(15%), 중반(70%), 후반(15%)으로 나누고, 각 구간별 목표(품질, 생산성, 향상 등)를 반영하여 보상 계산.

## 4. Conclusion

### 다목적 함수 결과

- 수학적인 최적함수를 바탕으로 각 운전 조건을 정량적으로 평가.
- 생산량이 +2.8% 증가하고, 수분율이 -0.64%. 후진기 압축률이 -6.67% 감소하여 품질 만족도와 에너지 효율의 균형 있는 향상을 달성.
- 제작 가능수 기반 전략이라는 면에서 실제 운전 기준으로 활용 가능.

### 강화학습 결과

- 공정 시간 구간별 목표에 맞는 자동으로 운전 전략을 도출.
- 수분율이 -1.72%, 후진기 압축률이 -24.24%로 더 큰 점 감안되며 에너지 효율과 품질 만족도 측면에서 우수한 성능을 보임. 생산량 향상은 +0.95% 수준으로 제작적이지만, 시계별 변화에 유연하게 반응한다는 면에서 동적이고 공정 대처 가능.

변수	전체 운전 시간 대비 (%)	구간별 (%)
생산량 (%)	438.71	401.99 (91.4%) 16.51 (3.9%) 15.70 (1.1%)
속도 (m/min)	753.81	750.41 (98.0%) 7.36 (0.9%) 2.23 (1.1%)
후진기 압축률 (%)	1.65	1.24 (75.0%) 0.44 (25.0%)
양질 (%)	161.65	166.93 (100.0%) (-1.27%)
제작 유량 (kg/min)	11.94	12.02 (100.0%) (-1.28%)
수분 (%)	4.46	4.43 (98.6%) (-1.72%)

## 5. Future Work

### 향후 연구 및 연구의 차별점

- 다목적 학습하는 경향을 가지고, 이를 바탕으로 운전 전략 수립에, 강화학습은 시계별 변화에 유연한 실시간 대응에 적용.
- 방향 두 방식을 통합 및 디지털 트윈과 연계하여 현장 중심의 최적 운전 시스템으로 확장 가능.

Q

캡스톤 디자인 II 중간발표

감사합니다

8

팀명 : EcoNOVA

팀원 : 박선아, 서지윤

지도교수 : 이상금

발표자: 서지윤

발표일: 2025.09.10

 국립한밭대학교

A