클라우드 서비스와 딥러닝을 활용한 재해방지 스마트 팩토리 시스템

SFT

대한설비관리학회 추계학술발표대회

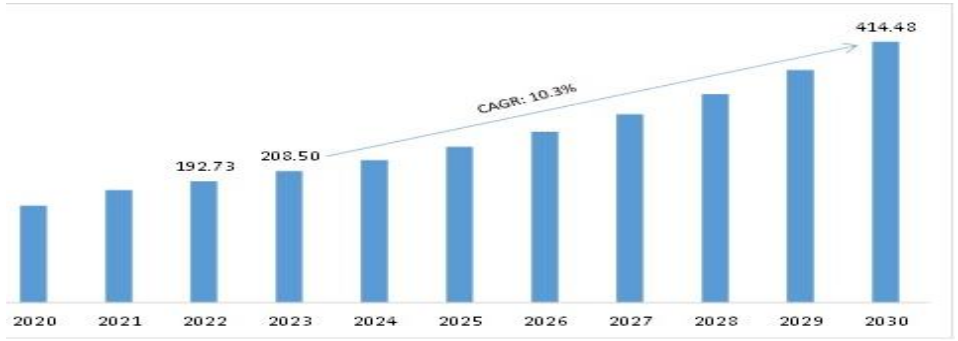
CONTENTS

01

연구 배경

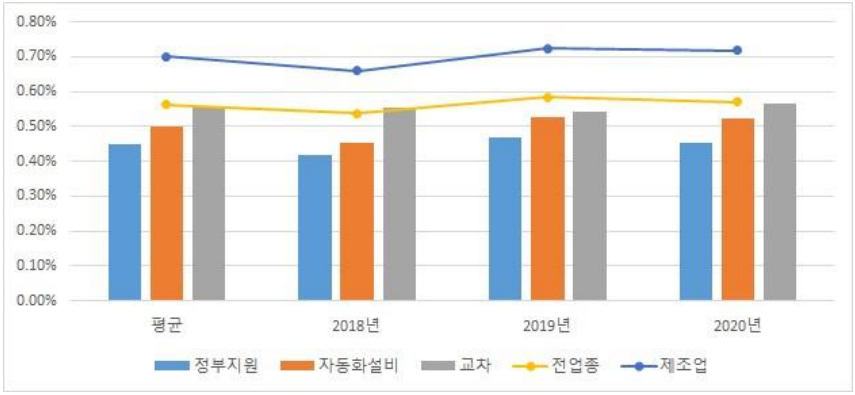
<스마트팩토리>

- 관리 비용, 인건비 최적화
- 원격 모니터링, 역량과 생산성에 대한 최적화
- 품질 향상 및 위험 예방, 인적 오류 제한



전 세계 스마트 팩토리 시장 규모는 연평균 10.3% 수준으로
성장할 것으로 예상

<스마트팩토리 재해율>



[2018 ~ 2020년도 스마트팩토리 사업장의 재해율 비교] [스마트팩토리 안전보건수준 파악 연구], 안전보건공단

국내 스마트팩토리의 빠른 구축·확산에 비해
유해·위험성은 크게 고려되지 않아 산업재해가 잦은 수준으로 발생

인천 서구 공장서 1.3t 장비에 깔려 50대 노동자 사망
2023.07.15 | 경기일보

현대차 울산공장서 30대 직원 기계 고장으로 숨져
2023.07.13 | 조선일보

고장으로 안 울렸다는 SPC 샤니공장 경고벨, 아예 없었다
2023.08.23 | 뉴스웰

제주 삼다수 공장서 고장 기계 수리 도중 끼어 숨져
2022.10..21 | 제주의소리

최근 장비 고장에 따른 인명피해 상당 수 발생

02

연구 목적

기존의 공정 과정

- ✓ 생산/품질 향상에 비해 고려되지 않는 안전율
- ✓ 사고나 재해 발생의 예측 및 방지의 어려움
- ✓ 장비 고장으로 인한 재해사고

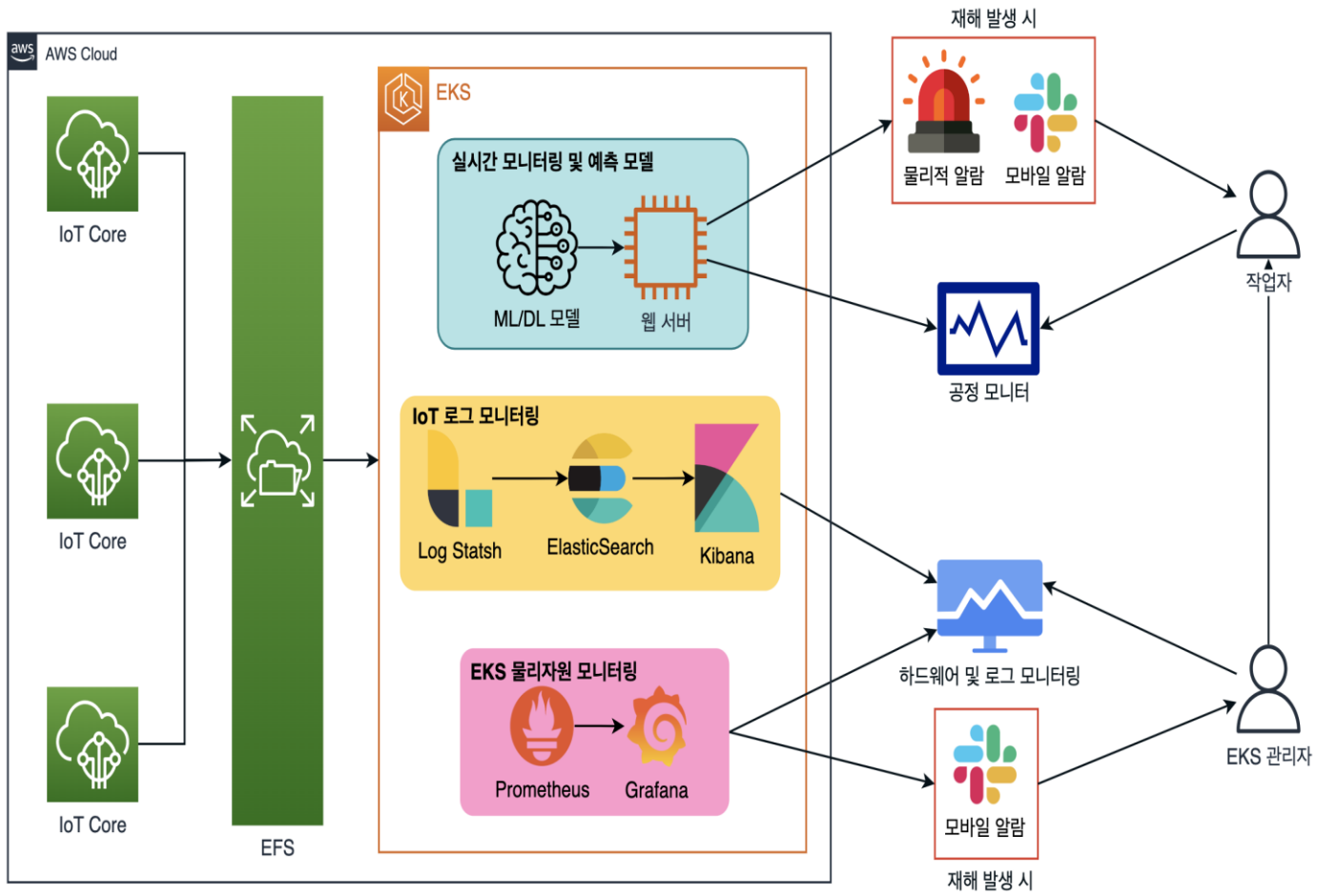
안전관리 스마트팩토리 DX화

- ✓ 실시간 모니터링/예측 모델
- ✓ 하드웨어 모니터링
- ✓ 로그 모니터링
- ✓ 재해 예방/방지 알림

기대효과

- ✓ 산업환경 작업자의 안전 관리
- ✓ 사고 및 재해로 인한 경제적 손실 방지
- ✓ 생산률/품질 향상
- ✓ 효율적 디버깅
- ✓ 공정 과정 안정화

03 시스템 흐름



<시스템 동작 설명>

1. **IoT Core** : 생산 라인의 실시간 데이터 수집.

2. **EFS** : 자동 확장 기능이 있는 볼륨.
   IoT Core에서 수집된 데이터 적재. (Database)

3. **EKS** : 모든 SW가 동작하는 가상 컴퓨터의 Cluster

4. **실시간 모니터링/예측 모델** :
   ML/DL 모델을 통해 데이터를 학습하고,
   실시간으로 생산라인의 데이터와 예측값 모니터링 제공

   >> 재해 상황 발생 시, 물리적 알람 및 모바일 알람 전송

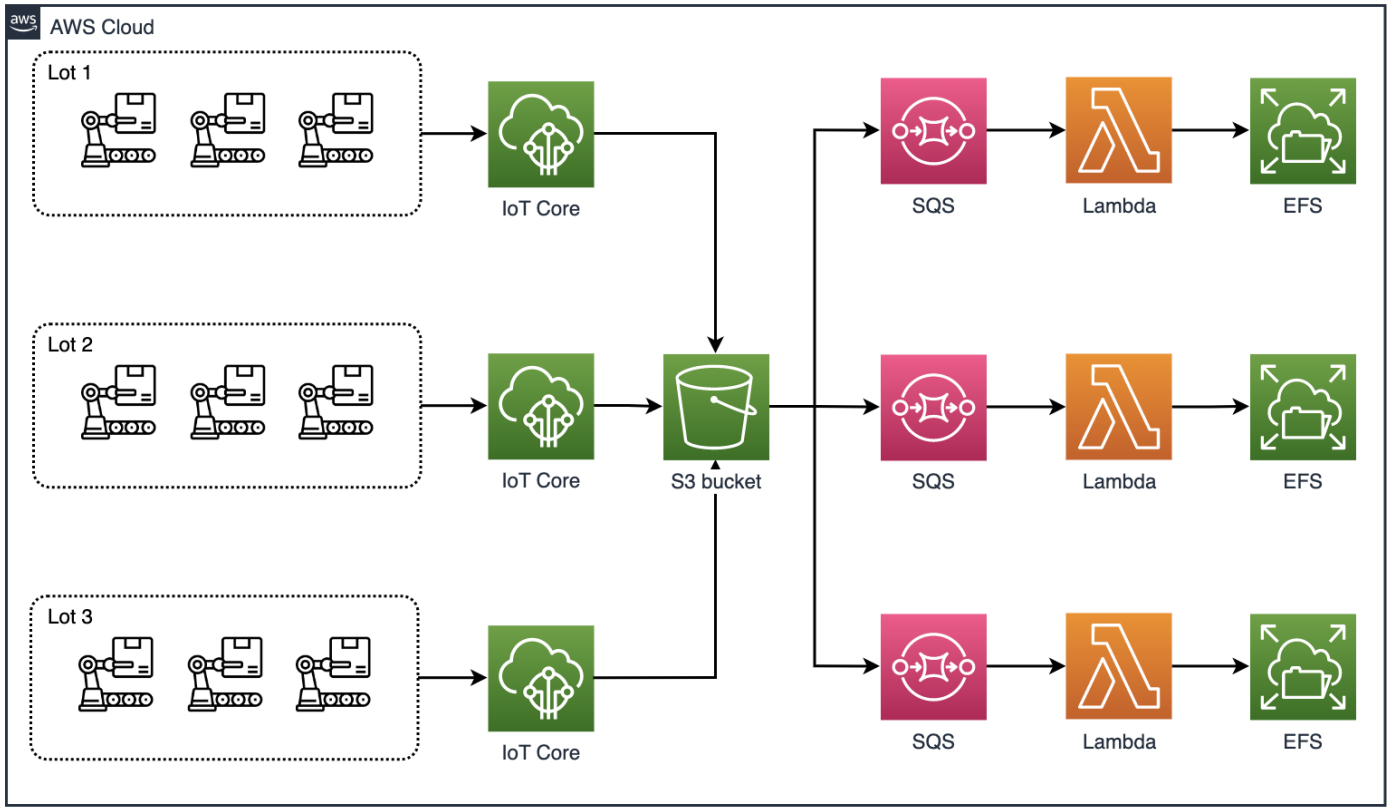
5. **IoT 로그 모니터링** : IoT Core 및 EKS 내부에서 동작하는 모든 SW들의 로그들을 수집한 UI제공

   >> Error Log를 필터링 하여 효율적인 디버깅 제공

6. **EKS 물리자원 모니터링** : EKS의 물리자원의 Metric을 수집 및 모니터링 제공

   >> EKS에 문제 발생 시, 관리자에게 알람을 전송/조치

03 데이터 수집 과정



1. LoT별 공정 기기와 IoT Core를 연동하여 데이터 수집

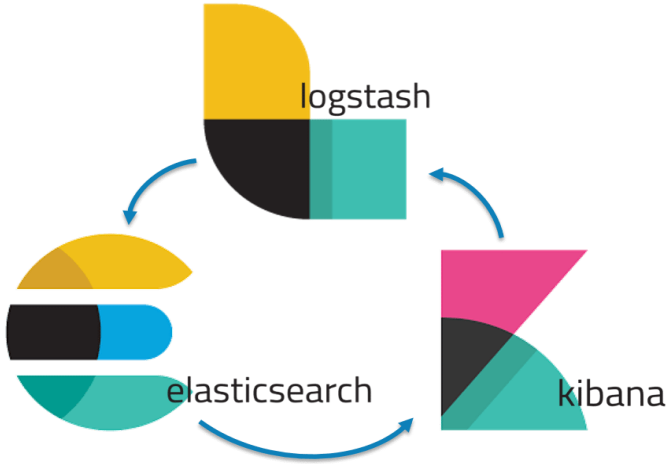
2. 대용량 객체 스토리지 S3 bucket에 모든 데이터 적재
   >> 타 AWS 클라우드 서비스와의 연동성 확보

3. Lot별 데이터를 분리하여 SQS에 메시지 형태로 전송
   >> 데이터 수집 과정의 안정성 증가

4. 적재된 메시지를 Lambda 함수를 이용하여 EFS에 전송
   >> 데이터의 실시간 전처리 가능

03 로그 및 하드웨어 모니터링

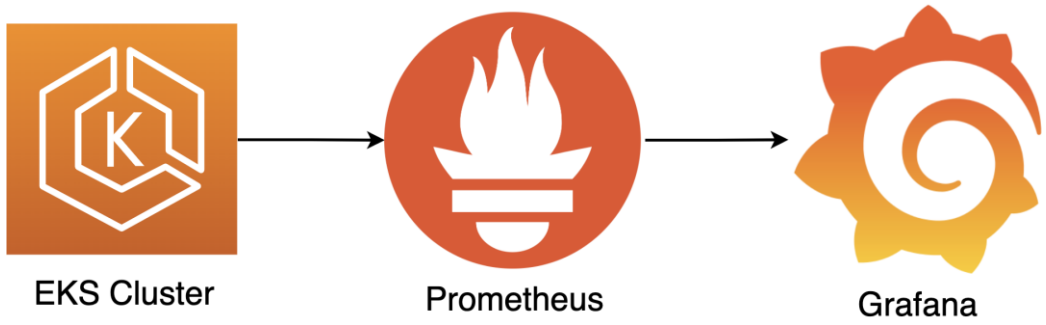
<로그 모니터링 – ELK Stack>



- Logstash : 로그 수집 엔진
  >> Cluster에서 동작하는 모든 SW 및 IoT의 로그 수집
- ElasticSearch : 검색 및 분석 엔진
  >> CNI를 통한 스토리지 동적 확장 지원
- Kibana : 시각화 및 탐색 도구
  >> Role-base 기반의 액세스 제한

>> 즉, 로그 수집 실시간 모니터링을 제공

<하드웨어 모니터링 – Kube Prometheus Stack>



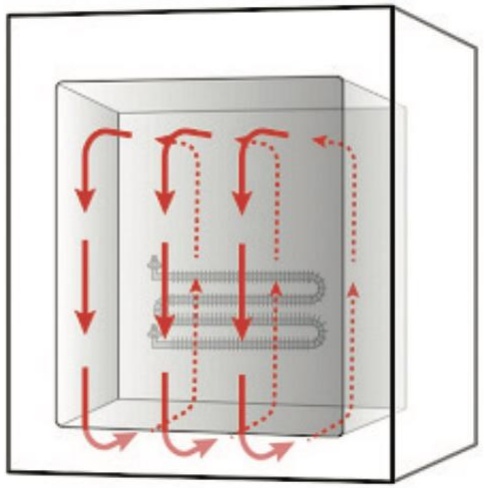
- Prometheus : 오픈소스 시스템 모니터링 및 경코 툴킷
  >> EKS Cluster의 HW Metric 수집
  >> 설정 임계값 이상의 Metric 수집 시, 알람 발송
  >> 자체 대시보드가 있으나, 가시성 및 확장성의 부족으로 Grafana와 연동
- Grafana : 최적 대시보드를 제공하는 오픈소스
  >> 시각화할 Metric을 CPU, Memory, Network, Volume 사용량으로 설정

>>즉, 하드웨어 사용량을 실시간 모니터링으로 제공

03

데이터 수집

<열처리 공정 소개>



- 도금처리 이후 피막과 안정제가 금속 표면에 안정적으로 달라붙을 수 있게 하는 공정
- 제품의 수명에 장기적으로 영향을 끼치는 공정.
- 공정으로 발생한 열기로 인해 작업자가 설비이상검사를 진행하기 어려움
- 가동 에너지 밀도가 높은 편으로 이로 인한 고장 多

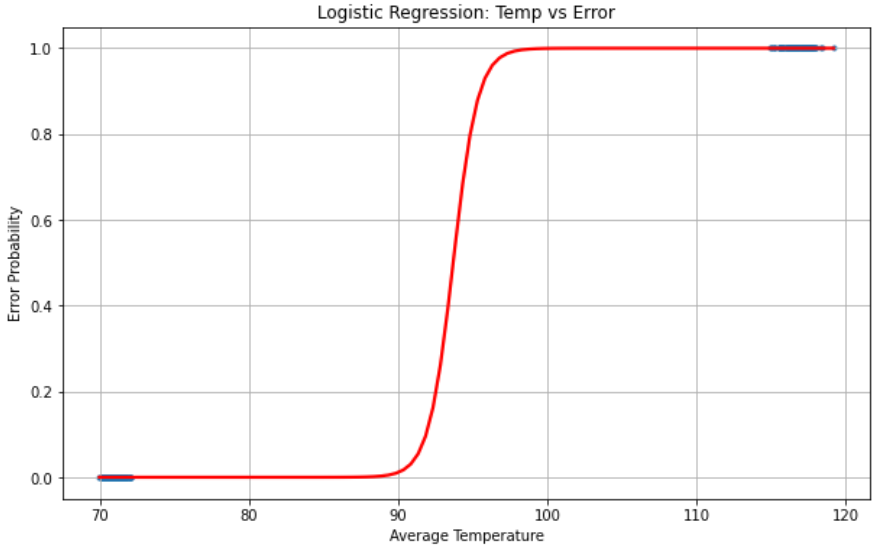
<현장 데이터 수집>

	process	date	temp_avg	current_avg	error
0	1	2021-09-06	70.95460	1.60835	FALSE
1	2	2021-09-06	70.99923	1.60789	FALSE
2	3	2021-09-06	70.57949	1.59961	FALSE
3	4	2021-09-06	71.20545	1.61122	FALSE
4	5	2021-09-06	71.12508	1.60954	FALSE
5	6	2021-09-06	70.41362	1.59656	FALSE
6	7	2021-09-06	70.77350	1.60011	FALSE
7	8	2021-09-06	71.03827	1.60759	FALSE
8	9	2021-09-06	70.84494	1.60511	FALSE
9	10	2021-09-06	70.78293	1.60361	FALSE
10	11	2021-09-06	71.08293	1.60522	FALSE
11	12	2021-09-06	71.41742	1.61779	FALSE
12	13	2021-09-06	70.54335	1.60017	FALSE
13	14	2021-09-06	70.85615	1.60066	FALSE
14	15	2021-09-06	70.76553	1.60139	FALSE
15	16	2021-09-06	70.88075	1.60463	FALSE
16	17	2021-09-06	70.90185	1.60636	FALSE
17	18	2021-09-06	71.20430	1.61034	FALSE
18	19	2021-09-06	70.89925	1.60728	FALSE
19	20	2021-09-06	115.26561	1.06002	TRUE
20	21	2021-09-06	116.73898	1.08680	TRUE

- 열처리 공정에서 수집된 온도 데이터와 고장여부
- 데이터를 바탕으로 시뮬레이션을 통하여 현장 재현
- 현장 상황에 따른 실시간 분석과 솔루션 제공

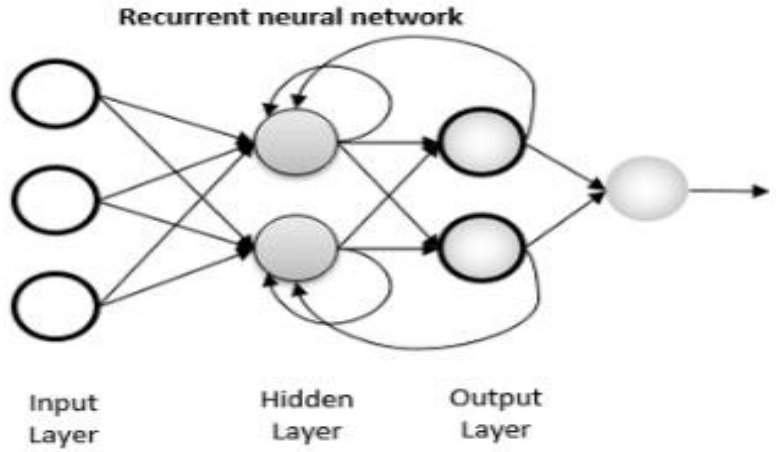
03 예측 및 실시간 모니터링 모델

<온도에 따른 고장률>



- 열처리 장비의 온도에 따른 불량률 회귀분석
- 온도와 고장율의 높은 상관성
- 시계열 온도 데이터를 바탕으로 고장 예측 모델 학습/검증 개발

<RNN 기반 예측 모델 학습>



- RNN은 각 시간 스텝에서 두 가지 입력을 받음
  >> 현재의 입력 값과 이전 시간 스텝의 hidden state
- 이 두 값을 결합하여 현재 시간 스텝의 출력과 새로운 hidden state 계산
- 이 새로운 hidden state는 다음 시간 스텝에서 사용되며, 이 과정이 시퀀스의 모든 원소에 대해 반복

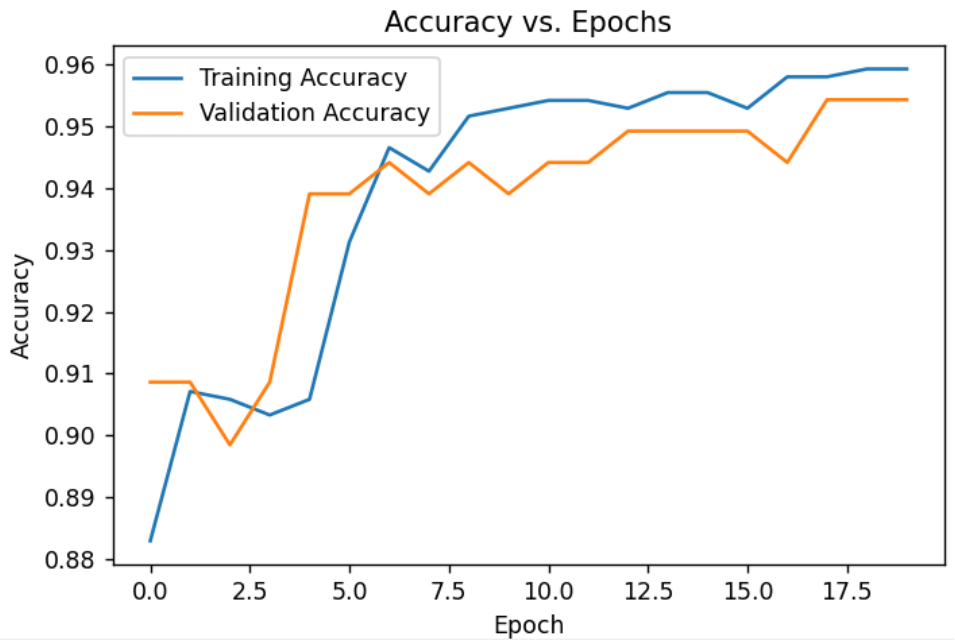
03 예측 및 실시간 모니터링 모델

<모델 학습 소스코드 - LSTM>

```python
# Scale the features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Split data into train and test sets with 70-30 ratio
X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.3, shuffle=False)

# Define a function to create sequences
def create_sequences(data, target, sequence_length):
    sequences = []
    target_values = []

    data = np.array(data)
    target = np.array(target)

    for i in range(len(data) - sequence_length):
        sequences.append(data[i:i + sequence_length])
        target_values.append(target[i + sequence_length])

    return np.array(sequences), np.array(target_values)

# Create sequences
sequence_length = 10
X_train_seq, y_train_seq = create_sequences(X_train, y_train, sequence_length)
X_test_seq, y_test_seq = create_sequences(X_test, y_test, sequence_length)

# Define model
model = Sequential([
    LSTM(50, activation='relu', input_shape=(X_train_seq.shape[1], X_train_seq.shape[2])),
    Dense(1, activation='sigmoid')
])

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(
    X_train_seq, y_train_seq,
    epochs=20,
    batch_size=32,
    validation_split=0.2  # Use 20% of the training data for validation
)

# Evaluate the model
loss, accuracy = model.evaluate(X_test_seq, y_test_seq)
```

- RNN 기반의 LSTM 모델로 시계열 데이터 학습
- 시퀀스에서 장기 패턴을 학습하는 데 유용

<모델 학습 결과>



- Accuracy : 0.9712
- loss : 0.0923
- Epoch : 50

>> 예측치 = 97.12%
>> 이를 바탕으로 실시간 예측모델 제공

04

결과

1. 실시간 모니터링 및 예측 모델

05

결론

스마트팩토리 DX화

데이터 학습분석을 통한 실시간 모니터링/예측모델 구현

>>RNN 기반 예측모델

하드웨어 및 로그 모니터링

>> IoT로그/EKS 물리자원 모니터링

>> 효율적 디버깅 환경제공 및 알림 시스템 구축

재해예방/방지 알림

>> EKS내 실시간 모니터링, 재해 발생 시 물리적 알람 및 모바일 알람 전송

기대효과 및 활용방안

산업환경 작업자의 안전관리

역량/생산성 최적화

사고 및 재해로 인한 경제적 손실 방지

효율적 디버깅

공정 과정 안정화

미래산업 현장에서의 안정된 생산자동화 시스템 구축

공장 DX화, 자동화 시스템을 통해 지능형 생산현장 발전에 기여