JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 10, pp. 2951-2960, Oct. 2024

Check for updates

AWS 클라우드 기반 신뢰성 있는 화재 감지 서비스 플랫폼 개발

최 환 석[1] · 이 건 희[2] · 조 한 성[2] · 최 민 석[2] · 이 우 섭[3*]
[1]경남대학교 게임학과 조교수
[2]국립한밭대학교 정보통신공학과 학사
[3]국립한밭대학교 지능미디어공학과 교수

Developing a Trusted Fire Detection Service Platform Based on the AWS Cloud

Hoan-Suk Choi[1] · Geon-Hee Lee[2] · Han-Seong Cho[2] · Min-Seok Choi[2] · Woo-Seop Rhee[3*]

[1]Assistant Professor, Department of Game Studies, Kyungnam University, Changwon 51767, Korea
[2]Bachelor's degree, Department of Information Communication Engineering, Hanbat National University, Daejeon 34158, Korea
[3]Professor, Department of Intelligent Media Engineering, Hanbat National University, Daejeon 34158, Korea

[요 약]

화재 발생시 적절한 초기 대응은 매우 중요하다. 2021년 쿠팡 물류창고 화재 사고에서는 6차례 화재 경보가 울렸지만 오작동으로 판단한 관리자가 화재 경보를 해제하는 바람에 수천억의 재산 피해가 발생했다. 이는 화재 감지기의 신뢰성 문제가 심각하다는 의미이다. 본 논문은 AWS 클라우드 환경을 기반으로 신뢰성 있는 IoT 화재 감지 서비스 플랫폼을 제안한다. 이는 센서 클러스터를 통해 데이터를 수집하고, 딥러닝 모델로 초기 판정후 앙상블 기법으로 화재를 최종 판정한다. 화재 발생시 시스템은 안전 관리자에게 알림을 보내고, 디지털 트윈과 실시간 영상을 통해 신속히 대응하게 한다. 모든 상황은 로그로 기록되어 분석 및 대책 수립에 활용되며 클라우드, 센서 퓨전, 딥러닝 기술을 융합하여 화재 감지 및 관리 정확성을 향상시킬 것이다.

[Abstract]

A proper initial response to a fire is very important. In the 2021 Coupang warehouse fire, the fire alarm sounded six times, but it was turned off by managers who thought it was a malfunction. The fire caused property damage valued at hundreds of billions of won. The malfunctioning of the fire detector caused a serious loss of trust in the fire detection platform. This paper proposes a reliable IoT fire detection service platform based on the AWS cloud environment, which collects field data through sensor clusters, builds a deep learning model to perform preliminary fire determination, and finally utilizes the Ensemble Method to determine the fire. When a fire occurs, the proposed system will notify safety managers, and they will be able to understand the situation through a digital twin and real-time video for a quick response. All situations are recorded in a log, which can be used for cause analysis and countermeasures. The proposed platform can greatly improve the accuracy and efficiency of fire detection, response, and management by converging cloud, sensor fusion, and deep learning technologies.

색인어 : 화재감지, 클라우드컴퓨팅, 디지털 트윈, 사물인터넷, 딥러닝
Keyword : Fire Detection, Cloud Computing, Digital Twin, Internet of Things, Deep Learning

http://dx.doi.org/10.9728/dcs.2024.25.10.2951

Received 31 July 2024; **Revised** 05 September 2024
Accepted 26 September 2024

***Corresponding Author; Woo-Seop Rhee**

Tel: +82-42-821-1749
E-mail: wsrhee@hanbat.ac.kr

Ⅰ. 서 론

화재로 인한 피해를 최소화하고 안전한 생활 환경을 조성하기 위해서는 초기에 화재를 감지하고 대응하는 것이 무엇보다 중요하다. 2021년 발생한 이천 덕평 쿠팡 물류창고 화재에서 6차례나 화재 경보가 울렸지만 관계자들은 오작동으로 판단하여 화재 경보를 해제했고, 두 명의 사상자와 수천억의 재산 피해가 발생했다[1]. 화재 감지기의 오작동은 감지 서비스 플랫폼 전반에 대한 신뢰도 저하와 인명, 재산 손실을 초래하며 안전에 대한 불감증을 유발한다. 따라서 본 논문에서는 정확하고 신속한 화재 감지를 통해 기존 화재 감지기의 신뢰도 및 안정성을 보완하기 위해 AWS 기반 화재 감지 서비스 플랫폼을 제안한다.

본 논문은 화재 감지를 위한 신뢰적인 IoT 서비스 제공을 위해 클라우드 기반 기계학습 기술인 AWS SageMaker를 활용하여 실시간으로 화재를 감지하는 AWS 기반 화재 감지 서비스 플랫폼을 설계하고 클라우드 기반 디지털 트윈 서비스인 AWS IoT TwinMaker를 활용하여 물리적으로 존재하는 실존 공간의 디지털 트윈을 구축하여 기존의 신뢰도가 낮은 화재 감지기를 보완하고 인명 및 재산 피해를 최소화하고자 한다. 또한 AWS IoT GreenGrass를 활용하여 IoT 디바이스를 효율적으로 관리하고, AWS IoT Sitewise를 통해 다양한 센서 데이터를 실시간으로 모니터링하고 수집한다. 센서 데이터는 온도, 습도, 일산화탄소, 카메라를 통한 현장 영상 데이터 등이 포함되며 이를 기반으로 AWS SageMaker를 활용해 OneClassSVM (One Class Support Vector Machine), YOLOv8 기반 머신 러닝 모델을 구축하였다. 이 모델을 통해 최종적으로 화재, 비화재가 판단되고, 최종 판단된 결과는 관리자에게 실시간 알람으로 전송된다. 화재 알람을 받은 관리자는 현장 상황을 실시간으로 원격 모니터링할 수 있는 관리자 모니터링 페이지를 통해 상황을 파악하고 화재 직후 신속하게 상황에 대처할 수 있다.

이를 위해 본 논문의 2장에서는 플랫폼 구축에 활용된 AWS IoT GreenGrass, AWS SageMaker, AWS IoT Sitewise, AWS IoT Twinmaker 등 관련 클라우드 기술을 설명한다. 3장에서는 제안하는 플랫폼의 기능 구현을 위한 클라우드 내의 서비스 구조, 센서 클러스터 하드웨어 및 화재 판정 알고리즘을 위한 딥러닝 모델 학습 및 최종 판단 모델에 대해 기술한다. 또한, 4장에서는 플랫폼 구현 결과로써 화재 판정 결과를 표현하기 위한 디지털 트윈과 관리자 모니터링 웹앱 개발 결과를 보이고, 5장에서 결론을 맺는다.

Ⅱ. AWS 클라우드 서비스

2-1 클라우드 기반 IoT 플랫폼, AWS IoT GreenGrass

대표적인 클라우드 기반 IoT 플랫폼인 AWS IoT GreenGrass는 IoT 애플리케이션 구축, 배포 및 관리 기능을 제공하는 클라우드 서비스이다[2]. 사용자는 AWS IoT GreenGrass를 사용하여 학습된 기계학습 모델을 배포하고, 디바이스가 생성하는 데이터를 통해 로컬에서 추론을 처리하며, 디바이스 데이터를 필터링 및 관리할 수 있다. 또한 로컬 네트워크의 다른 디바이스 뿐 아니라 AWS IoT 코어와 안전한 통신수단을 제공하여, IoT 데이터를 안전하게 클라우드로 전달할 수 있다. AWS IoT GreenGrass는 구성 요소(Component)라고 하는 소프트웨어 모듈을 기반으로 에지 애플리케이션을 구축할 수 있으며 에지 디바이스를 AWS 서비스 또는 타사 서비스와 쉽게 연결할 수 있다[3].

2-2 클라우드 기반 기계학습 서비스, AWS SageMaker

AWS SageMaker는 완전 관리형 기계학습 서비스로 머신러닝 모델을 쉽고 빠르게 구축하고 학습시킬 수 있으며, 학습된 모델을 프로덕션 지원 호스팅 환경에 직접 배포할 수 있다[4]. Jupyter Notebook 인스턴스를 제공하여 탐색 및 분석에 필요한 원본 데이터의 쉬운 접근을 지원하며 분산된 환경 내 대규모 데이터를 효율적으로 실행하는 데 최적화된 다양한 머신러닝 알고리즘을 제공한다. 또한 특정 워크플로우에 맞게 조정되는 유연한 분산형 학습 기능을 제공하며 클릭 몇 번으로 모델을 학습하고 확장 가능한 모델 배포를 지원한다[5].

2-3 클라우드 기반 산업용 IoT 플랫폼, AWS IoT Sitewise

AWS IoT Sitewise는 클라우드 기반 산업용 IoT 플랫폼으로 산업장비 데이터 수집, 모델링, 구성 및 분석 등의 기능을 제공한다. 사용자는 AWS IoT Core의 MQTT 메시지를 라우팅하거나 OPC-UA(Unified Open Platform Communications Architecture) 프로토콜 등을 통하여 데이터를 AWS IoT Sitewise에 업로드 할 수 있다. 이렇게 수집된 데이터는 모델링 단계를 거친 뒤 체계적으로 정리된다. 실시간 데이터로 구성된 자산(asset)들은 실시간으로 대시보드를 통해 시각화가 가능하여 데이터를 이해하고 분석하는데 도움이 된다[6].

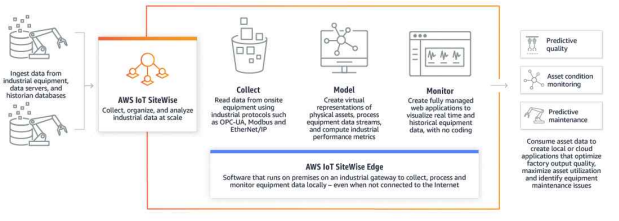


그림 1. AWS IoT Sitewise 아키텍처
Fig. 1. AWS IoT Sitewise architecture

2-4 클라우드 기반 디지털 트윈 서비스, AWS IoT TwinMaker

디지털 트윈 (Digital Twin)이란 물리 시스템의 데이터를 기반으로 가상환경 내 구현된 물리 객체의 디지털 표현이다. AWS에서는 디지털 트윈 서비스인 AWS IoT TwinMaker를 제공한다. 이는 다양한 센서, 카메라 등을 이용하여 물리적 공간을 가상환경에 구현하여 실제 데이터를 사용해 모니터링하고 오류 진단, 수정 및 최적화가 가능하다[7]. TwinMaker의 Component는 AWS IoT SiteWise, Amazon Kinesis Video Streams, Amazon Simple Storage Service(S3) 기반 기본 제공 데이터 커넥터를 통하여 서비스의 데이터 저장소에 접근할 수 있다. Scene은 디지털 트윈에 나타내는 가상환경이며 이는 Resource로 구성되어 TwinMaker 리소스 라이브러리를 통하여 jpg, glb, glTF같은 3D 모델들을 업로드한 뒤 Scene편집기를 통하여 디지털 트윈을 구성할 수 있다.

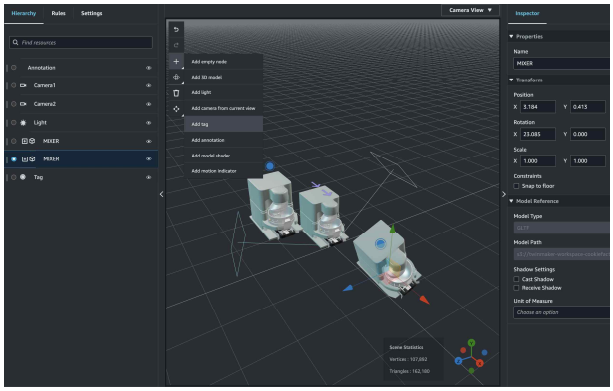


그림 2. AWS IoT Twinmaker Scene 편집기
Fig. 2. AWS IoT Twinmaker Scene editor

Ⅲ. 제안하는 AWS 기반 화재 탐지 서비스 플랫폼

3-1 제안하는 서비스 플랫폼 구조

본 논문에서 제안하는 화재 판정 알고리즘과 플랫폼은 AWS 클라우드에서 동작한다. 이는 네트워크만 있다면 공간 제한 없이 서비스 플랫폼을 구축할 수 있다는 것을 뜻한다. 이는 AWS 내에서 데이터 처리와 화재 판정이 이루어지고, 대부분의 과정이 동작하기 때문에 제공하는 서비스에 대한 안정성과 효율성을 보장받는다.

본 서비스 플랫폼을 구성하기 위하여 AWS IoT Core, AWS IoT SiteWise, AWS ECR, AWS SageMaker, AWS TwinMaker, AWS Grafana 서비스를 사용하였다. AWS는 여러 서비스들 간에 유기적인 연결을 지원하고, 기타 외부 프레임워크나 서비스도 AWS와 연결되어 서비스 플랫폼을 더욱 효율적으로 구성할 수 있다.

그림 3은 제안하는 화재 감지 서비스 플랫폼의 구조를 나타낸다. 우선, 라즈베리파이에 부착된 센서를 통해 수집되는 데이터는 MQTT를 통해 AWS IoT GreenGrass 메시지로 캐스팅된다. 메시지들은 AWS IoT Core 서비스에 등록된 후 AWS IoT Sitewise의 센서 별 Asset 데이터로 수신된다. 수신된 데이터는 여러 AWS 서비스로 전달된다.

먼저, Lambda로 전달된 데이터는 화재 판정을 위한 데이터로 사용된다. 이때 웹앱 모니터링 페이지로의 데이터 전달을 위해 AWS ECR과 AWS API Gateway가 사용되고, AWS SNS는 관리자 알림 전송을 위해 활용된다.

Kinesis Video Streams로 전달된 데이터는 타겟 공간 실시간 카메라로도 사용된다. 카메라를 통해 실시간 수집되는 데이터가 실시간 타겟 공간 카메라이자 화재 판정을 위한 영상 입력 데이터로 사용되는 것이다. 이때 사용되는 프로토콜은 영상 특화 프로토콜인 RTSP (Real Time Streaming Protocol)이다.

또한 Sitewise의 수집 데이터는 SageMaker를 통해 기 훈련된 모델로 전달된다. 모델은 화재 여부를 판단하고 다른 Lambda 함수를 통해 판단 결과에 대한 트리거를 발생시켜 Vue.js로 구성되고 EC2 서버를 통해 구동되는 관리자 모니터링 페이지로 전달된다. 모니터링 페이지을 구성하기 위한 데이터 시각화와 디지털 트윈 구축은 각각 AWS IoT TwinMaker와 AWS managed Grafana가 담당한다[8].

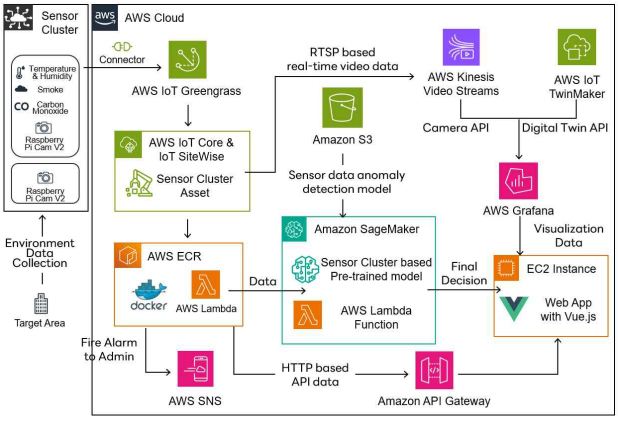


그림 3. 제안하는 서비스 플랫폼 구조
Fig. 3. Proposed service platform structure

3-2 센서 클러스터 구성 환경

제안하는 플랫폼은 GreenGrass 코어 디바이스로 라즈베리파이 4B[9] 2대를 사용하였고, DHT22 (온, 습도 센서), MLX90614 (적외선 온도 센서), MQ-7 (일산화탄소 센서) 및 라즈베리파이 카메라 모듈, Intel RealSense[10] 카메라를 그림 4와 같이 설치하고 Python3.7 버전을 사용해 코드를 작성하여 IoT 환경을 구축하였다.

센서들과 Intel RealSense에 부착되어 있는 1번 라즈베리파이는 센서 데이터를 수집 및 처리하고 카메라를 통해 시각

데이터를 처리한다. 라즈베리파이 카메라 모듈이 부착되어 있는 2번 라즈베리파이는 관리자 모니터링 페이지에 실시간 카메라를 임베딩하기 위해 사용된다. 관리자 모니터링 페이지에서 카메라 패널 또한 AWS를 통해 임베딩되기 때문에 1번과 2번 라즈베리파이 모두 코어 디바이스로 구성하였다.

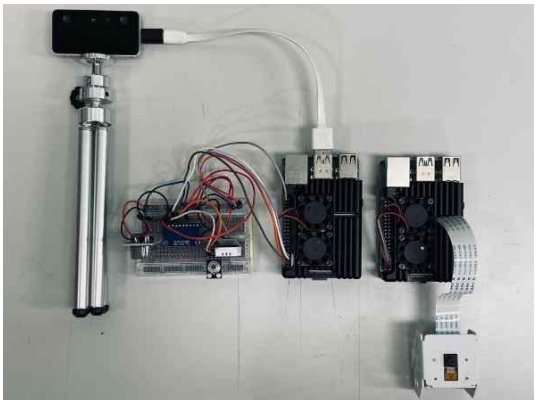


그림 4. 센서 클러스터 구성
Fig. 4. Sensor cluster configuration

3-3 화재 판정 딥러닝 모델 구축

1) 판정 모델 구축을 위한 데이터 수집

센서 클러스터로 수집된 데이터들은 AWS의 관계형 데이터베이스 서비스인 AWS RDS (Relational Database Service)[11]에 업로드 되어 .csv 파일로 구성하였다. 모델을 점진적으로 학습시켜 판정 결과를 고도화하기 위해 단기간에 수집된 데이터가 아닌 여름철, 겨울철 별로 수집 기간을 나눠 데이터를 학습시켰다. 또한 초기 데이터, 중기 데이터, 최종 데이터 별로 데이터를 학습시켜, 기간 별 학습 데이터에 대한 지속적인 성능 판정 및 과적합 방지 작업을 거쳤다. 이를 위해 여름철을 6월부터 8월로 설정하였고, 겨울철은 10월부터 11월까지 설정하여 2주 기간 동안 센서 데이터를 수집했다. 또한 모든 데이터는 2주간 24시간 종일 수집되어 시간별 센서 데이터의 변화를 모델에 학습할 수 있도록 하였고, 센서 클러스터를 외부와 내부의 경계인 창문에 두어 내부 온도 및 외부 온도 변화 데이터를 모두 수집하도록 했다.

파일 내부의 릴레이션은 온도, 습도, 일산화탄소, 적외선 온도, 적외선 습도를 속성으로 가지고, 튜플은 2초 딜레이를 가지는 시계열 데이터이다. 여름철과 겨울철 일정 기간 내 데이터를 수집하고 이를 합산하여 하나의 파일로 만든 결과, 모델 학습을 위해 개별 속성(온도, 습도, 일산화탄소, 적외선 온도, 적외선 습도)을 가지는 튜플(시계열 데이터)은 총 53만개로 구성되었다.

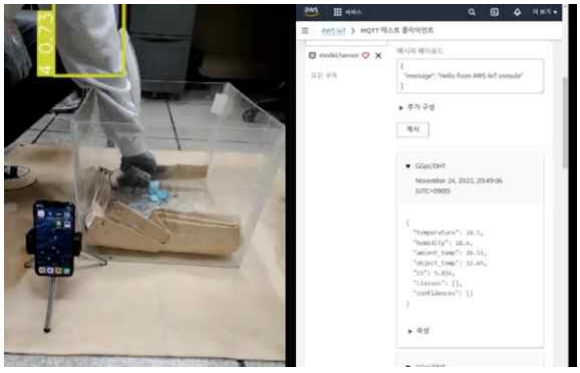
2) 실시간 개별 센서 데이터 기반 1차 화재 판정

OneClassSVM (One-Class Support Vector Machine)을 통해 구축된 센서 데이터 화재 판정 모델은 센서별로 화재를 판정한다. 이는 Anomaly Detection, 즉, 이상 상황을 감지하기 위함이다[12]. 각 센서를 통해 실시간으로 감지되는 온도, 습도, 적외선 온습도, 일산화탄소 등의 데이터는 기훈련된 모델을 통해 화재를 1차적으로 판정한다. 이를 위해 53만개의 주변 환경 센서 데이터를 학습시킨다.

이 데이터들은 OneClassSVM 기반 모델에 학습되어 속성별로 개별 이상 상황 탐지 모델을 생성한다. 이 모델은 센서 클러스터에서 실시간으로 수집되는 주변 환경 데이터에 대해 현재 상황이 이상 상황인지 정상 상황인지 판정하게 된다.

1차 화재 판정 모델의 성능을 고도화하기 위해 특정 기간 데이터 분할 학습, 점진적 데이터 학습 등의 방법을 적용했다. 또한 새로운 데이터를 학습시킬 때마다 모델 성능 판정을 진행하여 모델 성능을 향상시킬 수 있으며, 실제 화재 실험을 통해 수집한 화재 상황 데이터와 화재 상황 가정 시나리오 별 데이터를 통해 정성적인 성능 분석을 진행하였다. 이는 실험실 레벨에서 그림 5와 같이 안전 박스를 설치하고 발화체(고체연료)를 투입하여 30회 반복 수행하였다.

우선 목표 학습 데이터를 여름철과 겨울철 데이터로 설정하고 초기 여름철 데이터를 수집하였다. 초기 데이터는 대략 6만 튜플로, 이 초기 데이터를 학습시킨 1차 판정 모델의 성능은 좋지 않았다. 결과치 변동이 상당히 심했고, 무엇보다도 겨울철 데이터가 학습되지 않았다 보니 영하 기온 데이터에 대해 화재로 판정하는 등 신뢰도가 매우 낮은 모습을 보였다. 이후 점진적으로 여름철 데이터를 모두 학습 시켰을 때 과적합 등의 부작용이 나타나지 않아 겨울철 데이터의 경우에도 여름철 데이터와 동일하게 점진적으로 학습시켰다. 최종적으로 53만개의 모든 데이터가 학습되었고, 가상 시나리오 상황 및 실제 화재 상황에 대해서도 좋은 성능을 보이는 1차 판정 모델이 구축되었다.



*screen shot of app operation.

그림 5. 화재 실험 환경
Fig. 5. Fire experiments environment

3) 최종 화재 판정을 위한 Ensemble Method

'예'와 '아니오'로 이진 판정된 1차 판정 값은 Hard-Voting, Soft Voting, Stacking 등의 방법으로 세분화되어 있는 앙상블 기법(Ensemble Method) 중 소프트 보팅을 통해 화재 여

부를 최종 판정한다[13].

소프트 보팅은 분류기(Classifier)의 레이블 값 결정 확률을 모두 더하고 평균을 구한 뒤 이들 중 확률이 가장 높은 레이블 값을 최종 보팅 결과로 선정하는 방법이다. 이때 분류기는 각 센서 별 판정 데이터에 해당한다. OneClassSVM을 통해 이진 값으로 판정된 센서 데이터들은 각 판정에 대한 확률값을 가진다. 이 확률은 평균을 도출하기 위한 입력값이 된다.

평균값이 도출된 후 가장 확률이 높은 레이블 값이 최종 결과가 된다. 만약 가장 확률이 높은 레이블의 값이 -1, 즉 화재라면, 최종화재 판정을 하고 화재 상황에 맞은 이후 대처가 실행된다. 이 시점에서 현재 상황을 정상 상황이라 판정한 센서 데이터가 존재한다면 그 센서에 대해 유지 보수 작업을 실행한다. 만약 가장 확률이 높은 레이블이 0, 즉 비화재 또는 정상 상황이라면 최종 화재 판정은 비화재가 된다.

3-4 AWS SageMaker을 통한 화재 판정 모델 학습

Amazon SageMaker는 AWS에서 제공하는 클라우드 기반의 기계 학습 서비스이다. SageMaker는 사용자가 모델을 구축하고 훈련할 수 있는 다양한 도구와 환경을 제공한다. 그 중 가장 대표적인 예로 Jupyter 노트북 인스턴스를 사용하여 데이터 탐색과 전처리 작업을 수행할 수 있으며, 내장 알고리즘 또는 사용자 정의 알고리즘을 사용하여 모델을 훈련할 수 있다. 또한 훈련 및 추론을 위해 필요한 컴퓨팅 자원을 자동으로 관리하며, 사용자는 필요한 자원 유형을 선택하기만 하면 되므로 매우 편리하게 모델 학습을 할 수 있다. 이처럼 Amazon SageMaker는 사용자 매우 간편하게 모델을 학습할 수 있는 환경을 제공하여 개발자는 개발 단계에만 집중할 수 있게 한다.

1) 1차 화재 판단 모델 구축을 위한 OneClassSVM

OneClassSVM은 비지도 학습 알고리즘으로 주로 이상 탐지에 활용된다. 이 알고리즘은 데이터셋에서 정상 데이터를 학습하여 새로운 데이터가 정상 데이터와 유사한지, 아니면 이상인지를 판별한다.

• 훈련 데이터

OneClassSVM은 단일 클래스에 대한 정보만을 가지고 훈련된다. 이상 데이터의 정보는 필요하지 않다.

그림 6의 데이터는 라즈베리파이의 부착된 여러 센서에서 정상 데이터를 수집한 .csv 파일 일부분이다. 데이터의 신뢰성을 얻기 위하여 여름, 겨울 계절과 실내에서 히터와 에어컨을 틀었을 때의 이상 상황으로 탐지하는 것을 방지하기 위해 히터와 에어컨을 가동하면서 모든 센서 데이터를 각각 2주 정도 수집하였다.

• 결정 경계

OneClassSVM 알고리즘은 훈련 데이터를 기반으로 결정

	A	B	C	D	E	F
1	data_id	temperatu	humidity	amb_temp	obj_temp	co
2	1	25.3	72.5	32.03	28.37	0.768
3	2	25.3	72.2	32.03	36.69	0.769
4	3	25.4	70.2	32.37	64.47	0.873
5	4	25.4	70.2	34.13	92.25	1.305
6	5	25.8	90.5	36.01	110.01	2.026
7	6	26.8	99.9	38.41	117.07	3.359
8	7	26.8	99.9	42.05	122.11	4.493
9	8	30.1	99.9	44.41	114.87	5.617
10	9	32.7	99.9	46.41	105.51	6.456
11	10	35	99.9	47.79	93.27	7.349
12	11	35	99.9	48.89	82.03	8.371

그림 6. 정상 데이터(.csv)
Fig. 6. Normal data(.csv)

경계를 형성하게 된다. 이 경계는 데이터의 분포를 나타내는 일종의 영역으로 훈련 데이터의 대부분이 이 영역 내에 위치하도록 한다.

OneClassSVM의 모델 학습시 nu 파라미터 값은 결정 경계를 결정하는 값으로, nu 파라미터 값이 작으면 작을수록 학습 데이터내에서 이상 탐지 경계가 더욱 좁아지고 nu 파라미터 값을 높일수록 경계가 넓어지게 된다. 즉, 민감한 탐지를 원할 경우 파리미터 값을 작게 조정하여 학습을 진행한다. 하지만 모델 학습은 사용자의 사용 요구나 상황에 맞게 유동적이여야 하며 이 과정은 파라미터를 반복적으로 조정하면서 최적의 값을 찾아내야 한다.

모델훈련 후 이상치 차단 테스트를 위해 임의의 값을 데이터 값으로 설정해 준 뒤 테스트를 진행한다. 이러한 작업을 통해 어느 범위까지 이상치로 판단하는지 찾아낼 수 있다.

2) YoloV8를 통한 영상 화재 감지

Object Detection 기법 중 하나인 YOLO (You Only Lock Once)는 CNN (Convolutional neural network)와 달리 이미지를 한장으로 해석한다. 또한 1-stage 방식으로 Classification과 Localization을 동시에 진행 하므로 정확도는 낮지만, 속도가 빠르다는 장점이 있다[14].

본 연구에서는 좌, 우 각도에서의 인식률이 떨어지는 기존 버전과 달리 새로운 리포지토리를 활용하는 Yolov8을 활용한다[15]. Yolov8 모델을 학습하기 위해 AWS SageMaker 인스턴스 노트북을 생성해 주어야한다. 정확도를 높이기 위해 학습 횟수를 높게 설정한다. 이때 이미지 학습의 경우 상대적으로 높은 수준의 리소스를 요구하기 때문에 인스턴스 유형을 'large'로 생성하는 것이 용이하다. 요구하는 리소스 대비 상대적으로 낮은 리소스를 사용할 경우 학습 진행 중에 인스턴스의 커널이 중단되는 현상이 발생할 수 있어 요구 리소스(GPU 자원)를 항상 고려하고 있어야 한다.

YOLO 모델과 관련된 Python 라이브러리로 Ultralytics 라이브러리를 사용한다. 이 라이브러리는 모델을 쉽게 사용하고 개선할 수 있는 도구를 제공한다. 학습 코드를 사용자가 직접 구성하는 것이 아닌 명령어 한 줄로 모델의 학습, 검증

까지 진행할 수 있도록 하는 라이브러리이며 개발자는 이미지 데이터 세트에만 중점을 두면 된다.

[!pip install ultralytics] 명령어를 통해 설치 할 수 있으며, 이 모델 학습 결과와 검증 결과는 yolo 디렉토리에 저장이 된다. YOLO(yolov8n.pt)는 설치한 ultralytics의 내장된 모델 정의 메서드이다. ultralytics에서 제공하는 기본 yolov8 모델을 설정하고 model.train의 파라미터 값으로 data = '데이터셋 디렉토리 위치', epochs = '학습 횟수', batch = '배치 사이즈', imgsz = '데이터셋의 이미지 사이즈', patience = '과적합 방지'와 같이 파라미터 값을 설정한다. 이후 커널을 실행하면 학습이 진행된다.

학습을 진행하게 되면 현재 진행되고 있는 학습 수, GPU 사용량, 손실률 등 정보가 표시된다. 또한 학습이 완료된 후 모델은 runs/detect/train2/weight/best.pt에 저장되며, 학습의 전반적인 결과를 확인 할 수 있다.

3) 최종 화재 판단 모델 구축

최종화재 판단 모델은 센서(온도, 습도, 일산화탄소, 적외선)의 Soft Voting 함수를 통해 도출된 판단 값과 yolo 알고리즘 기반으로 도출된 카메라 센서 데이터의 판단 값을 비교하여 최종 화재 판정을 내리게 된다. 카메라 모델을 Soft Voting에 포함시키지 않은 이유는 카메라 모델의 판단 결과 값이 다른 센서와 비교해 봤을 때 더 정확한 결과를 내리기 때문에 Soft Voting에 포함하지 않고 개별적으로 처리하였다.

센서 모델은 AWS S3 Bucket에 저장하여 활용할 수 있도록 하며, 해당 모델이 저장된 디렉토리 위치를 설정한다. 다음으로 lambda_handler event 파라미터로 전달받는 각각의 센서 데이터를 새로운 변수 설정을 진행한다. 여기서 event 파라미터는 AWS Lambda에서 설정한 AWS IoT Event 즉, 현재 센서 데이터를 가져오게 된다. 그 후 predict_sensor의 모델 디렉토리 위치와 실시간 데이터 값이 파라미터로 넘어가 실시간 데이터의 판단 값을 predict_sensor 함수가 반환하게 된다. 이와 마찬가지로 predict_accuracy 함수는 판단 값의 정확도를 반환하고 이 정확도 값은 sigmod 메서드를 통해서 0~1사이의 정규화된 값으로 설정되게 된다. 마지막으로 soft_voting 메서드를 통해 각 센서의 정확도 값은 하나의 최종 판단 값을 반환받게 된다. yolov8 모델 기반 예측값 img_class는 객체 탐지 클래스의 결과를 딕셔너리 형태로 초기화 되어있다. Soft_voting의 값과 img_class값을 비교하여 최종적으로 화재 및 비화재를 판단하게 된다.

Ⅳ. 제한하는 서비스 플랫폼 구현결과

4-1 AWS IoT TwinMaker을 이용한 디지털 트윈 구축

1) TwinMaker 작업공간 생성 및 Scene 설정

우선 TwinMaker 콘솔에서 작업 공간을 생성하여 작업 공간의 이름 및 디지털 트윈을 만들면서 저장되는 리소스 (3D 모델 등)가 들어갈 버킷을 지정한다.

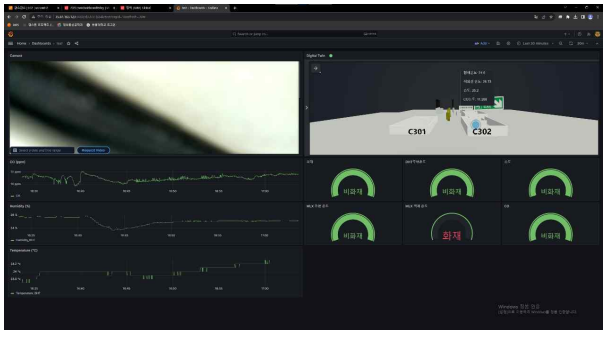
디지털 트윈을 시각화 하기 위해서는 서드 파티 애플리케이션으로 Grafana를 사용하여야 하는데 TwinMaker에서는 Amazon에서 제공하는 Grafana와 사용자 구동 Grafana 서버 중 하나를 선택할 수 있다. Amazon에서 제공하는 Grafana에서는 웹으로의 임베딩이 불가능하므로 본 시스템에서는 자체 관리형 Grafana를 선택하였다. 작업공간을 만든 후 리소스 라이브러리에서 리소스 추가를 통해 3D 모델을 업로드한다. 추가한 리소스들을 바탕으로 Scene을 하나 생성하면 그림 7과 AWS IoT TwinMaker Scene 편집기를 확인할 수 있다. 이 편집기를 통해 객체를 추가하고 실제 객체가 투영되는 디지털 트윈을 구성할 수 있다.

*screen shot of app operation.

그림 7. AWS IoT TwinMaker Scene 편집기
Fig. 7. AWS IoT TwinMaker Scene editor

2) AWS managed Grafana을 활용한 디지털 트윈 및 센서 데이터 시각화

최종적으로 구성한 디지털 트윈을 표현하기 위해 먼저 가상 컨테이너 서비스인 도커 (Docker)를 설치하고 Grafana 서버를 가동시킨다. Grafana 서버 생성 후 SiteWise와 TwinMaker의 데이터 소스를 추가하면 대시보드를 통해 시각화 할 수 있다. 우선 대시보드 및 패널을 생성한다. 패널에는 여러 종류가 있지만 본 논문에서는 시스템 구축에 4가지 유형의 패널 (디지털 트윈, 비디오 피드, 시계열 그래프, 게이지 차트)을 설계하였다.



*screen shot of app operation.

그림 8. Grafana 시각화 대시보드
Fig. 8. Grafana based vitalization dash board

• **시계열 데이터**

시계열 데이터는 대시보드의 쿼리 탭에서 SiteWise의 데이터 소스를 가져오게 한다. 그다음 Get property value history 쿼리 타입을 사용하여 특정 기간 동안 수집된 데이터를 가져오게 하고 측정값을 설정하면 데이터 시각화 그래프를 구성할 수 있다.

• **게이지 차트**

게이지 차트의 경우에도 시계열 데이터와 마찬가지로 동일하게 설정한 뒤, Get property value 쿼리 타입을 통해 특정 시간의 데이터를 가져오게 한다. 게이지의 모양은 사용자 맞춤으로 구성할 수 있다.

• **디지털 트윈**

디지털 트윈은 AWS IoT TwinMaker Scene Viewer를 활용하여 패널을 생성한다. 추가적으로 작업공간과 장면 등을 선택하면 이상 없이 설정된 상태일 경우 디지털 트윈이 나타난다.

• **비디오 피드**

비디오 피드의 경우 AWS IoT TwinMaker Video Player 패널을 선택한 뒤 카메라 스트리밍 정보가 담긴 SiteWise 자산을 선택하고, Kinesis Video Stream에서 실행되는 스트리밍 이름을 선택하면 비디오 피드가 생성된다.

위와 같이 4가지 유형의 패널을 추가하여 적절한 위치에 배치한다면 웹앱에서 그림 8과 같이 AWS managed Grafana를 활용한 디지털 트윈 및 센서 데이터 시각화 결과를 볼 수 있다.

4-2 화재 상황 감시를 위한 관리자 모니터링 웹앱 개발

본 서비스 플랫폼은 최종 화재 판정 시각화를 위해 Vue.js 프레임워크를 활용한 화재 감지 서비스 플랫폼의 모니터링을 위한 웹앱을 개발하였다. 이 페이지는 실시간 데이터 모니터링, 시스템 상태 확인, 경고 및 알림 기능을 제공하여 시스템 관리자가 효율적으로 화재 감지 서비스 플랫폼을 관리할 수 있도록 설계하였다. 이를 통해 서비스 플랫폼 관리자는 복잡한 관련 지식 없이도 화재 상황을 신속하게 파악하고 적절한 대응 조치를 취할 수 있도록 한다.

주요 정보가 한눈에 들어오는 대시보드, 쉽게 접근할 수 있는 메뉴 구조, 그리고 응답성이 뛰어난 디자인을 적용했다. AWS IoT SiteWise 및 AWS IoT TwinMaker에서 수집된 데이터는 Vue.js 웹앱 페이지에 통합되어 실시간으로 표시된다. 이를 통해 관리자는 센서의 현재 상태, 경고, 알림 등을 실시간으로 확인할 수 있다. 웹 페이지에는 다양한 정보를 보여주는 대시보드가 포함되어 있다. 이 대시보드는 화재 감지 서비스 플랫폼의 전반적인 상태, 센서 데이터의 통계, 그리고 이벤트 로그 등을 제공한다.

센서 클러스터에서 수집된 데이터는 웹앱 페이지에 실시간으로 업데이트되며, 서비스 플랫폼의 현재 상태를 즉각적으로 파악할 수 있다. 화재 감지 시, 웹 페이지의 주요 요소와 현재 상황 표시 위젯에 빨간색의 경고 색상을 적용하여 긴급 상황을 눈에 띄게 표시하였다. 이에 더해 최종 화재 판정 결과, 즉 현재 상태는 깜빡이는 애니메이션 효과를 추가하여 빠르게 사용자의 주의를 집중시키게 하였다. 본 기능을 구현하기 위해 CSS의 keyframes와 animation 속성을 사용하였다.

또한, 화재 여부는 디지털 트윈에 실시간 적용되어 관리자는 트윈을 통해 센서 클러스터가 설치된 공간 내 어느 공간에서 화재가 발생하였는지에 빠르게 알 수 있다. 이는 현재 상태 표시 위젯과 동일하게 관리자의 신속한 직후 상황 대처를 위해 고안되었다.

1) 관리자 모니터링 메인 페이지

모니터링 페이지에 접속하면 가장 처음 보이는 페이지인 메인 페이지는 타겟 공간, 즉 센서 클러스터가 설치된 공간 내 온도 및 습도, 일산화탄소 농도를 보여준다. 이 페이지는 그림 9와 같이 센서 클러스터가 감지하는 데이터를 숫자 및 그래프 형태로 실시간 시각화하여 관리자가 각 현재 공간의 상태를 즉시 파악할 수 있도록 설계되었다. 또한 이 페이지는 현재 공간의 상태가 안전한지, 아니면 화재 위험이 있는지를 한눈에 확인할 수 있는 대시보드를 제공한다.

*screen shot of app operation.

그림 9. 관리자 모니터링 페이지, 정상 및 화재 상황

Fig. 9. Administration monitoring page, normal / abnormal

2) 센서 데이터 페이지

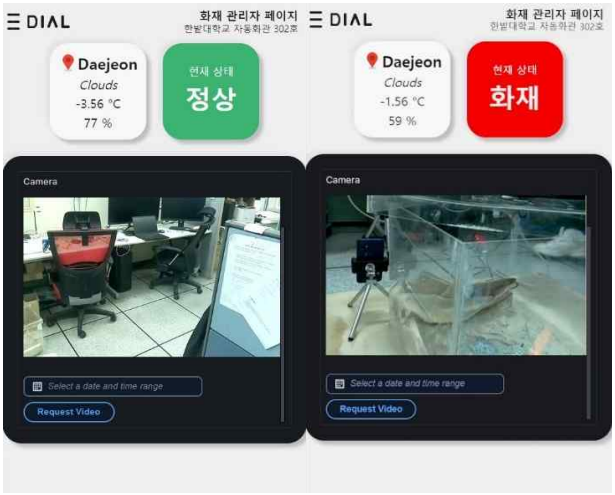
센서 데이터 페이지는 측정된 각 센서의 데이터를 보여준다. 이는 그림 10과 같이 온도, 습도, 일산화탄소 센서들의 판정 데이터를 시간별로 화재 및 정상으로 표시하여 센서의 화재 판정 상태를 파악할 수 있도록 하였다. 또한, 이 페이지에서는 각 센서의 상태를 실시간으로 확인하고, 센서가 정상적으로 작동하지 않는 경우 즉시 대응할 수 있도록 하였다.

*screen shot of app operation.

그림 10. 센서 판정 데이터 페이지, 정상 및 화재 상황
Fig. 10. sensor determination page, normal / abnormal

3) 실시간 카메라 페이지

실시간 카메라 페이지는 관리자의 위치에 상관없이 현장 상황을 알 수 있도록 하기 위해 구성되었다. 실시간 카메라는 센서 클러스터가 설치된 공간을 실시간으로 모니터링하며, 관리자는 이 페이지를 통해 실시간 영상을 통한 현장 상황을 확인할 수 있다. 이는 그림 11과 같이 화재 발생 시 실시간으로 현장 상황을 파악하고, 필요에 따라 적절한 대응 조치를 취할 수 있도록 돕는다.



*screen shot of app operation.

그림 11. 실시간 카메라 페이지, 정상 및 화재 상황
Fig. 11. Real-time camera page, normal / abnormal

4) 디지털 트윈 페이지

디지털 트윈 페이지는 관리자의 화재 현장 파악에 도움을 준다. 화재가 발생하면 센서 클러스터가 이를 확인하고 AWS TwinMaker에 이 사실이 전달된다. AWS TwinMaker는 이 데이터를 기반으로 디지털 트윈에 어느 공간에 화재가 발생했는지 시각화한다. 이 시각화 프로세스는 그림12와 같이 실시간으로 반영되어 관리자는 디지털 트윈을 통해 더욱 신속한 화재 상황 대처가 가능하다.

*screen shot of app operation.

그림 12. 디지털 트윈 페이지, 정상 및 화재 상황
Fig. 12. Digital twin page, normal / abnormal

V. 결 론

본 논문은 클라우드 환경의 기계학습 서비스 및 지능형 IoT 시스템을 기반으로 AWS 기반 화재 감지 서비스 플랫폼 개발에 대한 내용을 제안하였다.

본 논문에서 제안하고 구축한 서비스 플랫폼, AWS 기반 화재 감지 서비스 플랫폼은 센서 클러스터를 통해 현장의 데이터를 수집하고 수집된 데이터를 통해 OneClassSVM 기반 딥러닝 모델을 구축하고 각 센서 별 1차 화재 판정을 내린 뒤 1차 화재 판정을 앙상블 기법(Ensemble Methods) 중 소프트 보팅으로 통합하여 최종 화재 판정을 도출한다. 최종 화재 판정이 내려지면 즉시 안전 관리자에게 자동으로 연락되고 관리자가 119 신고, 대피 알람 등 화재 상황에 대해 원활하게 대처하게 된다.

따라서 본 서비스 플랫폼은 단순한 화재 감지기가 아닌 클라우드와 센서 퓨전, 딥러닝 등의 기술을 모두 융합한 구조로, 화재 감지 이후 안전 관리자 또는 시설 관리자의 사후 대처까지 가능하도록 설계되었다. 최종적으로 화재가 발생한 것으로 판정되면 즉시 안전 관리자에게 이 사실이 전달되고 관리자는 디지털 트윈 및 실시간 카메라를 통해 센서의 오류를 판단할 수 있도록 하고, 동시에 현장 상황을 파악하여 빠르게 대처할 수 있도록 하였다. 이 모든 기능은 아마존의 클라우드 서비스인 AWS를 통해 구현되기 때문에 AWS에서 보장하는 안정성과 효율성을 보장 받고, 모든 상황이 텍스트

로그 형태로 남아 화재 발생 후에도 로그를 통해 당시 상황에 대한 정보를 파악할 수 있다. 나아가 추후 화재 상황에 대한 대비 및 대책을 수립할 수 있다. 이를 기반으로 본 서비스 플랫폼은 앞으로의 화재 감지 및 대응에 있어 효율성과 정확성을 높이는데 이바지할 수 있을 것으로 생각된다. 하지만 특정 클라우드 서비스에 대한 의존성으로 인해 다른 클라우드 환경의 호환성을 저해할 수 있으며 이를 해결하기 위해서는 TwinMaker, Greengrass와 같은 특화 서비스를 오픈소스 기반 솔루션으로 대체하는 추가적인 연구가 필요할 것으로 생각된다. 또한 다른 딥러닝 모델이나 기법을 적용하여 신뢰성을 보강하는 추가 연구를 진행할 예정이다.

참고문헌

[1] Hankyoreh. Coupang Deliberately Switched off Emergency Bells 6 Times during Fire…3 Managers Charged [Internet]. Available: https://www.hani.co.kr/arti/area/capital/1004118.html.

[2] Amazon AWS Documentation. What Is AWS IoT Greengrass? [Internet]. Available: https://docs.aws.amazon.com/greengrass/v2/developerguide/what-is-iot-greengrass.html.

[3] Amazon AWS Documentation. How AWS IoT Greengrass Works [Internet]. Available: https://docs.aws.amazon.com/greengrass/v2/developerguide/how-it-works.html.

[4] Amazon AWS Documentation. What Is Amazon SageMaker? [Internet]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html.

[5] Amazon AWS Documentation. SageMaker Autopilot [Internet]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/autopilot-automate-model-development.html.

[6] Amazon Web Services. AWS IoT SiteWise [Internet]. Available: https://aws.amazon.com/ko/iot-sitewise/.

[7] Amazon AWS Documentation. AWS IoT TwinMaker? [Internet]. Available: https://docs.aws.amazon.com/iot-twinmaker/latest/guide/what-is-twinmaker.html.

[8] T. Abirami, S. Mapari, P. Jayadharshini, L. Krishnasamy, and R. R. Vigneshwaran, “Streamlined Deployment and Monitoring of Cloud-Native Applications on AWS with Kubernetes Prometheus Grafana,” in *Proceedings of 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, Faridabad, India, pp. 1149-1155, November 2023. https://doi.org/10.1109/ICAICCIT60255.2023.10465818

[9] Raspberry Pi. Raspberry Pi 4 Tech Specs [Internet]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.

[10] Intel. Intel RealSense™ ID Solution F455 [Internet]. Available: https://www.intel.co.kr/content/www/kr/ko/products/sku/212561/intel-realsense-id-solution-f455/specifications.html.

[11] Amazon AWS Documentation. What Is Amazon Relational Database Service (Amazon RDS)? [Internet]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html.

[12] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, “High-Dimensional and Large-Scale Anomaly Detection Using a Linear One-Class SVM with Deep Learning,” *Pattern Recognition*, Vol. 58, pp. 121-134, October 2016. https://doi.org/10.1016/j.patcog.2016.03.028

[13] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, “A Survey on Ensemble Learning,” *Frontiers of Computer Science,* Vol. 14, pp. 241-258, April 2020. https://doi.org/10.1007/s11704-019-8208-z

[14] GitHub. Ultralytics YOLOv8 [Internet]. Available: https://github.com/ultralytics/ultralytics.

[15] K. Geum, J. Park, C. Lee, H. Choi, Y. Go, and D. Kim, “Fire Detection and Notification System Using YOLO Object Detection Technology,” in *Proceedings of the 2024 KIIT Summer Conference*, Jeju, pp. 267-270, May 2024.

최환석(Hoan-Suk Choi)

2009년 : 한밭대학교 멀티미디어공학과 (공학사)

2011년 : 한밭대학교 멀티미디어공학과 (공학석사)

2018년 : 한밭대학교 멀티미디어공학과 (공학박사)

2018년～2020년: 한밭대학교 멀티미디어공학과 박사후연구원

2020년～2021년: 한국과학기술원 전기및전자공학부 위촉연구원

2021년: 주식회사 토브데이터 매니저

2021년～2024년: 카이스트-메가존클라우드 지능형 클라우드 융합기술 연구센터 매니저

2015년～현 재: 한국 ITU연구위원회 국제표준전문가

2024년～현 재: 경남대학교 게임학과 조교수

※관심분야 : IoT, Social IoT, Semantic Processing, Trust management, Trust Chain, 개인정보보호, GDPR, Cloud Computing, 스마트 제조, Generative AI

이건희(Geon-Hee Lee)

2024년 : 한밭대학교 정보통신공학과 (공학사)

2023년～2024년: 한밭대학교 정보통신공학과 학부연구생
2024년～현 재: ㈜비전세미콘 소프트웨어팀 사원
※관심분야 : Cloud Computing, IoT, 응용 S/W, 하드웨어 제어

조한성(Han-Seong Cho)

2024년 : 한밭대학교 정보통신공학과 (공학사)

2023년: 한밭대학교 정보통신공학과 학부연구생
※관심분야 : Cloud Computing, Deep Learning, Google Cloud, Amazon Web Service

최민석(Min-Seok Choi)

2024년 : 한밭대학교 정보통신공학과 (공학사)

2023년: 한밭대학교 정보통신공학과 학부연구생
※관심분야 : Cloud Computing, Computer Vision, Deep Learning

이우섭(Woo-Seop Rhee)

1983년 : 홍익대학교 (공학사)
1995년 : 충남대학교 (공학석사)
2003년 : 충남대학교 (공학박사)

1983년～2005년: 한국전자통신연구원 팀장/책임연구원
2012년～2013년: 프랑스 Institute TelecomSudParis 방문교수
2018년～2019년: 영국 Liverpool John Moores University 방문교수
2005년～현 재: 한밭대학교 지능미디어공학과 교수
2006년～현 재: 한국ITU연구위원회 국제표준전문가
※관심분야 : Semantic Processing, Trust Management, Trust chain, IoT, Social IoT