

Implementation of a Wi-Fi Mesh-based Fire Detection System using Multiple Sensor Nodes

Cha-Hun Park*, Hyoun-Chul Choi**, Myeong-Chul Park*

*Professor, Dept. of Avionics Engineering, Kyungwoon University, Gumi, Korea

**CEO, DAON Co., LTD, Gumi, Korea

[Abstract]

In this paper, we propose a Wi-Fi Mesh-based fire detection system for fire detection and rapid response. Previous fire detectors had the problem that it is difficult to distinguish between fire and non-fire based on a single sensor, and since multiple detectors operate independently, there is a lack of interconnectivity. In this paper, we present a fire detection system based on a K-NN classification model using a multi-sensor based fire detector. Also, by constructing a mesh network for fire detection, detectors within a spatial range can be interlinked to detect fire. Looking at the performance evaluation results of the implemented system, it was confirmed that the TPR(True Positive Rate) of fire classification was 96.1%, the FPR(False Positive Rate) was 0%, and the F1-Score, which corresponds to the harmonized mean value of accuracy and reproduction rate of fire and non-fire classification, was 98.01%, and the prediction accuracy ACC(Accuracy) showed excellent performance of 98.05%. In the future, we intend to develop it into an intelligent fire detector system through mesh network monitoring and multi-sensor self-diagnosis functions.

▶ **Key words:** Fire Detection System, Multi-Sensor, Wi-Fi Mesh Network, K-NN Classification Model

[요약]

본 논문은 화재 감지 및 신속한 대응을 위한 Wi-Fi Mesh 기반의 화재 감지 시스템을 제안한다. 기존 화재감지기는 단일 센서 기반으로 화재와 비화재에 대한 판별이 어렵고 다수의 감지기가 독립적으로 동작하기 때문에 상호 연계성이 부족한 문제점을 가진다. 본 논문에서는 다중 센서 기반의 화재감지기를 통하여 K-NN 분류 모델 기반의 화재 감지 시스템을 제시한다. 또한, 화재 감지용 메시 네트워크 구축을 통하여 공간 범위내에 있는 감지기가 상호 연계되어 화재를 감지할 수 있게 한다. 구현된 시스템의 성능평가 결과를 살펴보면, 화재 분류의 TPR(True Positive Rate)는 96.1%, FPR(False Positive Rate)는 0%, 화재와 비화재 분류의 정밀도와 재현율의 조화 평균값에 해당하는 F1-Score는 98.01%로 확인되었고 예측 정확도 ACC(Accuracy)는 98.05%의 우수한 성능을 보였다. 향후, 메시 네트워크 모니터링 및 다중 센서 자가진단 기능 등을 통하여 지능형 화재감지기 시스템으로 발전시키고자 한다.

▶ **주제어:** 화재 감지 시스템, 다중 센서, Wi-Fi 메시 네트워크, K-NN 분류 모델

- First Author: Cha-Hun Park, Corresponding Author: Myeong-Chul Park
- *Cha-Hun Park (chpark@ikw.ac.kr), Dept. of Avionics Engineering, Kyungwoon University
- **Hyoun-Chul Choi (daon@idaon.co.kr), DAON Co., LTD
- *Myeong-Chul Park (africa@ikw.ac.kr), Dept. of Avionics Engineering, Kyungwoon University
- Received: 2024. 09. 19, Revised: 2024. 10. 15, Accepted: 2024. 10. 24.

I. Introduction

화재감지기는 자동화재탐지설비로서 화재 발화 시점에서 발생하는 열, 연기, 불꽃 등을 감지하여 사이렌 등의 음향장치를 작동하여 화재 조기 대응을 가능하게 하는 설비로서 2017년 2월부터 ‘소방시설 설치 및 관리에 관한 법률’에 따라 모든 공동주택에 의무적으로 설치해야 한다. 하지만, 법률에는 화재감지기의 사용 연한이나 교체 시기에 관한 규정이 마련되지 않아 관리의 사각지대에 놓인 실정이다. 소방청에 따르면, 2021년부터 2023년 7월까지 자동화재탐지설비와 자동화재속보설비 등의 화재경보기 오작동으로 인한 소방차 출동 건수는 249,445건으로 전체 출동 건수의 96.6%에 해당한다. 이는 소방력 낭비와 출동으로 인한 화재 대비 공백 상태를 가중하는 원인으로 작용하고 있다. 더 심각한 문제는 실제 화재를 감지하지 못하는 오작동이 반대로 발생할 수 있다는 것이다. 한국소방산업기술원에 따르면, 10년이 지난 경보기는 정상적인 성능을 기대하기 어려우며 수도권 공동주택 화재경보기 800개를 점검한 결과, 83.93% 경보기가 오작동을 한 것으로 발표하였다. 기술원에서는 15년마다 화재경보기 교체를 의무화하기를 권고하지만, 경보기, 전선, 인건비 등의 천문학적 비용으로 현실적인 대안이 되지 못하고 있다[1]. 오작동의 주요인은 내구성 저하와 단일 센서를 통한 화재 식별, 노후화로 인한 감지 상태 악화 등이다. 또한, 화재 감지를 위한 공간 범위내에서 다수의 감지기가 독립적으로 동작하여 화재의 조기 인식하지 못하는 문제점을 가진다.

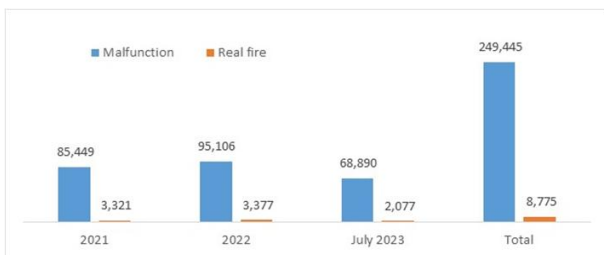


Fig. 1. Number of fire trucks dispatched due to fire alarm malfunction[1]

본 논문에서는 일산화탄소 감지, 유기 화합물(유해가스) 감지, LPG 및 LNG 감지, 온습도 감지, 연기 감지 등의 다중 센서를 보다 정확한 화재 여부를 식별하고, 화재감지기에 메시 네트워크 시스템 구축과 머신러닝(K-NN 분류 모델) 통하여 지능형 감지 시스템을 구현하고자 한다. 논문의 구성은 2장에서 기존 화재 감지 시스템 연구에 대해 살펴보고 3장에서 화재 감지 시스템의 구체적인 설계에 관해

기술한다. 4장에서 화재 감지 시스템의 실제 구현과 요소별 성능측정의 결과에 대해 상세히 설명하고 5장에서 결과에 관해 기술한다.

II. Background

1. Related works on Fire Detection System

머신러닝, 정보기술, 센서 및 신호 처리 기술의 발전으로 다양한 화재 감지 방법이 소개되고 있다. 또한, 지능형 처리 기술로 센서의 감지시간과 허위 경보를 줄이고 있으며, 시스템 성능 향상으로 실시간 비디오 기반 화재 감지 기술이 가능해지고 있다[2]. Nguyen(2023)은 원격 경보와 낮은 사양 컴퓨터 장치에 배포하기 위한 목적으로 경량 합성 신경망을 이용하여 96% 이상의 정확도를 보이는 화재 감시 시스템을 소개하였다[3].



Fig. 2. Fire Classification System(Left: On CPU, Right: On GPU)[3]

Ryu(2020)는 전처리 통하여 영상 내의 관심 영역을 검출하고 화재 여부를 합성곱 신경망을 이용해 추론하여 92.3%의 정확도와 93.5%의 정밀도로 화재를 검출하였다. 주요 특징은 화염이나 연기뿐만 아니라 연무 형태의 실내 연기 형성 여부도 판별할 수 있다는 것이다[4]. 또한, 기존의 낮은 검출 속도를 보완하기 위하여 딥러닝 모델을 이용한 연구도 소개되었는데, Bang(2023)은 YOLOv7을 이용하여 화재 영역을 검출하고 색상 필터를 적용하여 배경을 제거한 불꽃 후보 영역만을 대상으로 화재 여부를 확인하는 시스템을 제안하였다[5]. Kim(2020)은 화재 위치를 감지하기 위하여 딥러닝을 이용하여 비디오 기반의 화재 감지 시스템을 제안하여 화재 감지를 위한 센서와 비전 기반 기술이 시간 및 공간적 복잡도가 높아 사용에 제한적인 문제를 해결하고자 하였다[6]. Ruzaij(2024)은 화재를 조기에 감지하기 위한 휴대용 센서 노드를 제안하였는데, 온도 및 습도 센서의 데이터, 공기 중의 이산화탄소 가스, 휘발성 유기 화합물 및 질소 산화물의 수준을 측정하는 것 외에도 공기 중의 입자상 물질 PM1, PM2.5 및 PM10의 농

도를 측정하여 화재 경보 임계값을 결정하고 모니터링할 수 있게 하였다[7]. 하지만, 대부분의 연구는 이미지나 동영상 등을 대상으로 머신러닝 등의 지능형 감지 시스템에 국한되어 있으며, 실질적인 물리적 감지를 위한 센서 데이터의 인식 및 감지기 간의 상호 정보 교류에 관한 연구는 미약한 실정이다.

III. Design

1. Wi-Fi Mesh Network

제안하는 전체적인 화재 감지 네트워크의 연결 구조는 Fig. 3과 같이, 화재 감지 디바이스, 라우터, 서버로 구성되며 상호 간에 신호 및 데이터를 공유하며 각각 신호를 송수신하여 보가 빠른 신호의 전달을 통해 화재 알람 및 대피 경보등을 공유할 수 있도록 한다.

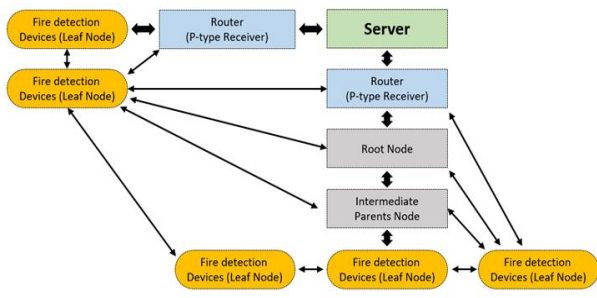


Fig. 3. Connection structure of a Wi-Fi Mesh-based fire detection network

네트워크는 다수의 노드로 구성된 LAN이며, 넓은 지역에 무선 노드를 이용하여 Wi-Fi 신호를 송출 및 수신하는 방식이다. 화재 감지 디바이스는 리프 노드로서 화재 감지 및 경보 알람을 제공하는 기기로 화재, 비화재, 평소 상황의 데이터를 직접 수집하여 송신하고 상위 데이터 저장 매체로서 데이터의 분석 및 판단을 수신하여 알람 및 경보를 제공할 수도 있다. 화재 감지 디바이스는 무선 또는 유선 네트워크를 통해 라우터로부터 직접 화재 감지 신호를 수신하도록 구성된다. 하위 노드들로부터 얻은 데이터를 수신 및 서버에 저장, 전달하고 수신기의 역할을 하며, 서버에서 전달받은 신호를 하위 노드에 송신하는 역할을 한다. 라우터는 논리적, 물리적으로 분리된 망을 지나는 패킷의 위치에 따라 최적화된 경로를 지정하는 기능을 수행하는 장비로서, 인터넷 공유기, 서버 등이 이에 해당하며, 본 연구에서 사용되는 라우터는 자동 화재 감지 시스템으로서 P형 수신기이다. 라우터가 있는 건물의 주소 정보와

해당 건물 내의 화재 발생 위치와 현재 화재의 정도와 진행 정보 등을 서버에 음성 신호로 혹은 텍스트(메시지)로 인코딩하여 전송 및 저장할 수 있다.

Wi-Fi 메시 네트워크는 Wi-Fi 신호가 도달할 수 없는 센서 디바이스 베로 시나리오에서 Wi-Fi 사각지대를 해소하고 영역별 안정적인 상태정보를 수집하기 위하여 사용된다[8]. 본 연구에서 ESP32-MeshKit를 활용하여 Mesh Network 구성하며 ESP32에서 제공하는 SDK 내부 컴포넌트를 이용한다[9]. ESP-WIFI-MESH 네트워크의 자동 라우팅 모드는 노드에서 보는 피어의 신호 강도 값에 따라 자동으로 형성되며 부모 노드가 꺼질 때마다 다른 노드 간의 자동 재연결을 용이하게 하여 메시 내에서 장애 안전 기능을 제공하는 특징이 있다. 또한, 표준 Wi-Fi 연결을 기반으로 하며 메시 노드 간 표준 WPA2 네트워크 보안을 사용하여 통신 보안을 보장한다. 사용되는 SDK의 주요 컴포넌트와 기능은 Table 1과 같다.

Table 1. Key components of ESP32 SDK

Component	Function
Creating a mesh network	Register one or more nodes, form connections between nodes, and set the name, security settings, and other components of the network
Register a node	Each node participating in a mesh network is given a unique identifier
Communication	Each node exchanges data with neighboring nodes and organizes data flow between nodes
Automatic routing	Determine the transmission path of data in a mesh network
security	Provides encryption and authentication functions to secure communication

2. Multi-sensor based sensing module

본 논문에서 구현한 화재 감지 시스템의 멀티 센서 기반 감지 모듈 테스트 회로는 Fig. 4와 같다. 화재를 감지하기 위한 다양한 방법이 있지만, 본 연구에서는 다중 센서 기반의 화재감지기를 통하여 더욱 신뢰성 있는 감지 기능을 제공하고자 한다. 또한, 수집된 다중 센서의 데이터를 근거로 머신러닝 기반의 화재 및 비화재 분류 기능을 탑재한다. 이를 위한 디바이스는 화재 발생 시 일산화탄소를 감지하는 센서, 포름알데히드 등 화재 시 발생하는 유해가스를 감지하는 휘발성 유기 화합물 감지 센서, LPG 및 LNG 감지를 위한 가스 감지 센서, 한정된 장소의 온도가 정해진 일정 온도가 되면 동작하는 정온식 열 감지를 위한 온습도 감지 센서, 빛을 이용하여 연기 감지를 위한 광전식 연기 감지 센서로 구성된다.

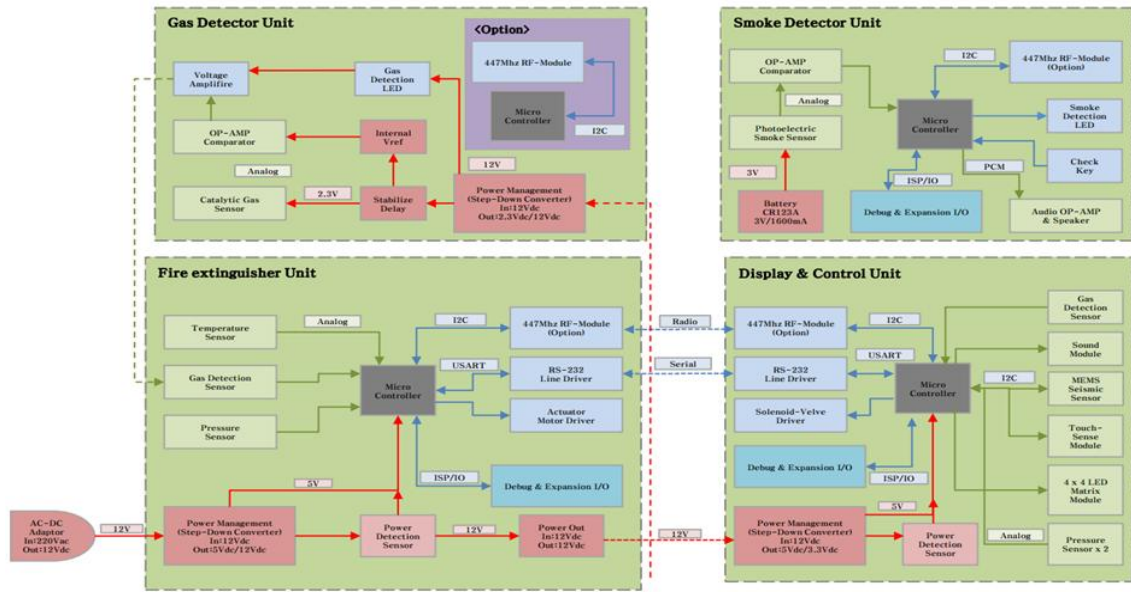


Fig. 4. Test circuit for multi-sensor based sensing modules

IV. Implementation

1. Implementation of Multi-sensor

구현된 멀티 센서 기반의 테스트 PCB와 구조는 Fig. 5 와 같다. 사용된 MCU는 Wi-Fi 와 블루투스를 내장하고 있는 Espressif System사의 ESP32-D0WD-V3 단일 칩 으로 Xtensa 32-bit 마이크로프로세서 외에 PWD, RTC, ROM, RAM 등이 내장되어 다양한 애플리케이션에 응용할 수 있는 장점이 있다. 일산화탄소를 감지하는 센서는 1000ppm까지 측정 가능한 ME2-CO 모델을 사용하였고 화재 발생 시 발생하는 유해가스(알코올, 담배 연기, 포름 알데히드, 벤젠)를 감지를 위한 센서는 검출 범위가 1~50ppm인 MP901 모델을 이용하였다. 또한, 가스 감지를 위해 접촉연소식 가스 센서인 MC105 모델을 이용하였고 온습도 감지는 SHT20 센서를 사용하여 설계하였으며 전체적인 구성 부품은 Table 2와 같다.

Table 2. Specification of multi-sensor detector

Item	Specifications
MCU	ESP32-D0WD-V3
CO sensor	ME2-CO
Gas sensor(LNG, LPG)	MC105
VOC Sensor	MP901
Hum. & Temp. sensor	SHT20

설계된 감지기는 연기 순환 농도 모의시험 결과, 연기가 들어와서 데이터가 특정될 때까지 최소 22초에서 최대 48 초가 소요되었다. 이는 Fig. 5와 같이 내부의 와류 동작으로 인하여 연기가 포화되는 속도에 개선이 필요한 것으로 분석되어 원형 구조와 챔버 돌출형태로 구조 개선이 필요하였다.

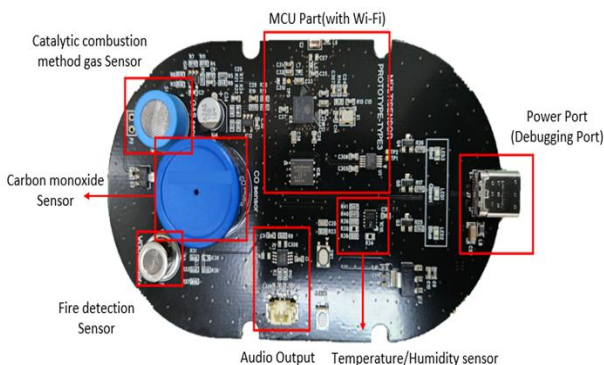


Fig. 5. Multi-sensor based Test PCB

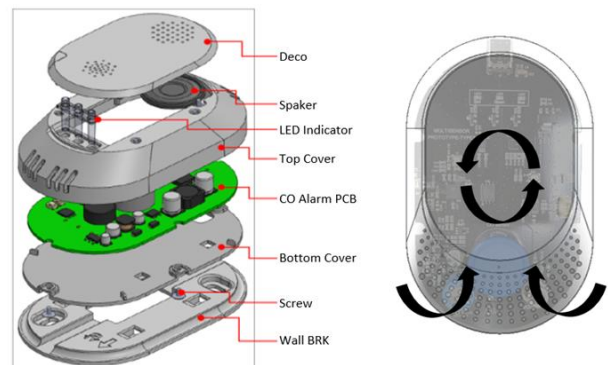


Fig. 6. Structure of Multi-sensor

Fig. 6에서 언급한 와류 현상을 개선하기 위하여 Fig. 7과 같이 구조 변경한 결과, 연기가 들어와서 데이터가 측정될 때까지 최소 5초에서 최대 13초로 개선되어 화재감지기 경보의 적합성을 도모하였다.

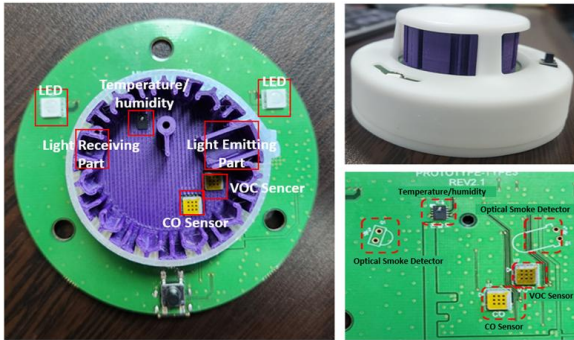


Fig. 7. Improved multi-sensor detector

2. Configuration of Wi-Fi Mesh Network

ESP32의 펌웨어 구성도와 ESP-WIFI-MESH의 다양한 시스템 이벤트에 대한 인터페이스는 Fig. 8과 같다. Wi-Fi 드라이버 위에 구축되며 루트 노드의 경우는 LwIP 스택을 사용할 수 있으며 각 애플리케이션은 Mesh Events 핸들러를 통해 네트워크와 인터페이스하고 ESP-WIFI-MESH는 Wi-Fi Stack 위에 구축되므로 애플리케이션이 Wi-Fi Event Task를 통해 Wi-Fi 드라이버와 인터페이스 할 수도 있다.

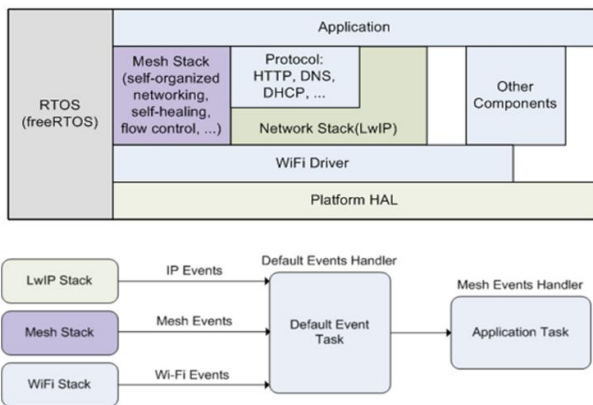


Fig. 8. ESP32 Firmware & System Event of ESP-WIFI-MESH

Mesh 네트워크 노드의 구성은 Fig. 9와 같이 세 가지 노드로 구분하여 구성된다. 각 노드는 상호 동등한 레벨의 스택을 가지고 있으며 레벨에 대한 영향이 없는 상호 동등한 관계를 가지지만 역할에 따라, Root Node, Intermediate Parent Node, Leaf Node로 나뉜다.

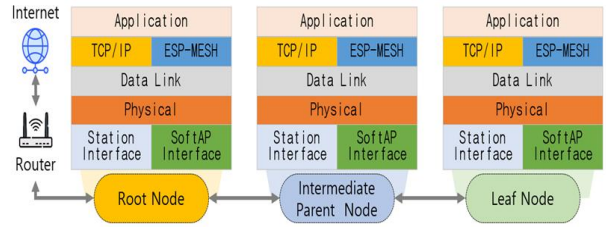


Fig. 9. Bidirectional Data Stream of ESP-WIFI-MESH

Root Node는 네트워크의 최상위 노드로서 전체 네트워크의 중앙 관리와 제어를 담당하며 Intermediate Parent Node는 중간 단계의 노드로 데이터의 중계와 라우팅을 수행하여 네트워크의 확장성과 효율성을 향상시키는 역할을 한다. 그리고 Leaf Node는 단말 단계의 노드로 데이터의 송수신지로 동작하여 실제 센서 데이터의 수집 및 제어를 담당한다. 그 외 노드로는 Idle Node가 있는데, 이 노드는 커버리지 영역에 신호가 도달하지 못한 노드로 판단되는 비활성 노드로서 네트워크 확장에 영향을 주지 않으며, 활성화되는 경우에도 이전 구성에 맞게 자동으로 연결 및 작동하게 된다. 노드 구성의 최대 허용 레이어는 3단계로 구성되며 네트워크의 확정성을 고려하여 제한되며 이를 통해 네트워크의 복잡성을 관리하고 효율적인 데이터 전달을 지원한다.

3. Experiment of Wi-Fi Mesh Network

Fig. 10은 구축된 Mesh 네트워크에 P형 수신기를 연동하여 동작하는 로그를 보인 것이다. Fig. 10에서 (a)는 Mesh Wi-Fi를 초기화하는 동작으로 흰색 로그는 menuconfig 설정에 따라 Wi-Fi를 구성한 것이고 녹색 로그는 Mesh가 시작되고 연관된 Task가 생성됨을 보인다. (b)는 menuconfig에 설정된 Wi-Fi를 찾을 수 없는 경우 연결 실패 로그를 보인 것이다. 흰색 로그는 Wi-Fi에 연결되지 않았다는 로그 표시이고 event 8은 root의 경우 부모 노드와 연결되지 않았다는 표시이다. (c)는 Wi-Fi 연결 성공한 경우이고 흰색 로그는 Mesh가 검색될 때까지 스캔을 반복하며 event 7은 부모 노드와 연결되었고 event 11은 root가 외부 IP 네트워크에 액세스할 수 있고 event 14는 root가 주소를 얻었음을 의미한다. (d)는 신규 자식 노드의 연결을 확인하는 로그로서 event 25는 연결된 노드들의 변동으로 라우팅 테이블이 변경되고 event 3은 신규 자식 노드가 연결됨을 의미한다. add_num은 이번엔 새로 생성된 자식 노드 수를 의미하고 total_num은 연결된 모든 자식 노드 수를 의미한다.

```
(a) Initializing Mesh Wi-Fi
I (1420) mesh: [CONFIG]connect to router:DAON_LAB_, 00:00:00:00:00:00
I (1432) [mwifi, 224]: MESH is started
I (1433) [mesh_wifi, 290]: event_loop_cb, event: 0
I (1433) [mesh_wifi, 294]: MESH is started
I (1438) [mesh_wifi, 219]: Note read task is running
I (1442) main: start state_task

(b) Wi-Fi connection failure
I (4278) [mwifi, 188]: Parent is disconnected, reason: 201
I (4279) [mesh_wifi, 290]: event_loop_cb, event: 8
I (4280) mesh: [wifi]disconnected reason:201(), continuous:1/max:12, root, v
ote(),<><>
I (4282) [mesh_wifi, 305]: Parent is disconnected on station interface

(c) Successful connection to Wi-Fi
I (1537) mesh: [scan]new scanning time:600ms, beacon interval:300ms
I (1543) [mwifi, 175]: Parent is connected
I (1546) [mesh_wifi, 290]: event_loop_cb, event: 7
I (1547) [mwifi, 258]: State represents: 0
I (1556) [mesh_wifi, 298]: Parent is connected on station interface
I (1557) mesh: <Flush>upstream packets, connections(max):6, waiting:0, upQ:0
I (1567) [mesh_wifi, 290]: event_loop_cb, event: 11
I (1575) [mesh_wifi, 290]: event_loop_cb, event: 14
I (1581) mesh: <Flush>root

(d) New child node connections
I (17222) [mesh_wifi, 290]: event_loop_cb, event: 25
I (17241) [mwifi, 243]: Routing table is changed by adding newly joined chil
dren add_num: 1, total_num: 4
I (17243) [mesh_wifi, 290]: event_loop_cb, event: 5
I (17247) [mesh_wifi, 310]: total_num: 4
I (17252) [mesh_wifi, 290]: event_loop_cb, event: 3

(e) Get an IP address
I (2154) [mesh_wifi, 290]: event_loop_cb, event: 65
I (2157) [mesh_wifi, 314]: Root obtains the IP address. It is posted by LwIP
stack automatically
I (2167) [mesh_wifi, 167]: TCP client write task is running
I (2168) [mesh_wifi, 79]: TCP client read task is running
I (2179) [mesh_wifi, 39]: Create a tcp client, ip: 192.168.0.168, port: 8107

(f) TCP socket connection failure
I (5178) [mesh_wifi, 39]: Create a tcp client, ip: 192.168.0.219, port: 8107
W (5394) [mesh_wifi, 54]: socket connect, ret: -1, ip: 192.168.0.219, port:
8107

(g) TCP socket connection successful
D (2688227) [mesh_wifi, 174]: ROOT read, size: 46, data: {S,N,N,30:c6:f7:56:
da:3c,3.0.0.1.63,29.9,27.7}
I (2688228) [mesh_wifi, 188]: S N N
D (2688231) [mesh_wifi, 197]: TCP write, size: 46, data: {S,N,N,30:c6:f7:56:
da:3c,3.0.0.1.63,29.9,27.7}
D (2690591) [mesh_wifi, 93]: TCP read, 54, size: 1456, data: {"group":"01:00
:5e:ae:ae:ae","data":"N"}
D (2690617) [mesh_wifi, 231]: Node receive: c8:2b:96:8c:a6:5c, size: 3, data
: "N"

(h) System information display
I (21442) [mesh_wifi, 256]: System information, channel: 10, layer: 1, self
mac: c8:2b:96:8c:a6:5c, parent bssid: 00:00:00:00:00:00, parent rssi: -120,
node num: 4, free heap: 175996
I (21461) [mesh_wifi, 259]: Child mac: 30:c6:f7:56:dc:78
I (21467) [mesh_wifi, 259]: Child mac: 30:c6:f7:56:da:3c
I (21473) [mesh_wifi, 259]: Child mac: 30:c6:f7:56:dc:80
```

Fig. 10. Mesh network log information linked to a P-type receiver

Fig. 10에서 (e)는 root가 IP 주소를 획득하는 로그를 보인 것으로 IP는 root만 얻을 수 있다. event 65와 66은 root가 IP 주소를 얻고 잃은 것을 각각 보인 것이고 root가 IP를 얻으면 TCP 통신 Task가 생성되고(TCP client task) menuconfig에 설정된 IP와 Port 번호로 TCP 소켓이 생성됨(Create a TCP client)을 확인할 수 있다. (f)는 menuconfig에 설정된 TCP 서버를 찾을 수 없는 경우에 소켓 연결 실패를 보인 것이고 (g)는 소켓 연결에 성공한 로그를 보인 것이다. 노드가 송신한 데이터를 root가 수신하고(ROOT read) 수신한 노드 데이터를 TCP 서버로 송신(TCP write)하며 TCP 서버가 송신한 데이터를 root가 수신(TCP read)한다. 마지막으로 (h)는 시스템 정보를 표시하는 것으로 주기적으로 Mesh 시스템 정보를 출력한다. 출력되는 정보는 현재 사용 중인 채널(channel), Mesh에서 root와의 연결 단계(layer), 자신에게 할당된 주소(self mac), 부모의 주소(parent bssid), 부모와 연결된 강도(parent rssi), 현재 연결된 Mesh의 노드 수(node num), 현재 자신에게 연결된 자식 노드의 주소(child mac)이다. 여기서, 연결 강도는 거리 및 장애물의 유무에 따라 연결 강도가 약해지면 값이 작아진다.

4. Implementation & Experiment of K-NN

화재감지기로부터 수집된 센서 정보를 이용하여 화재와 비화재를 분류하기 위하여 단순하면서도 예측력이 높은 K-NN(K-Nearest Neighbors) 분류 알고리즘을 이용하였다[10]. 먼저, 화재 데이터 수집은 나무, 종이, 플라스틱(ABS) 발화를 통해 데이터를 수집하고 비화재 데이터는 일반상태, 물, 음식조리, 담배연기를 통하여 학습 데이터를 수집하였다. Fig. 11은 데이터 수집 및 실험을 위한 챔버의 내부 환경을 보인 것이다.

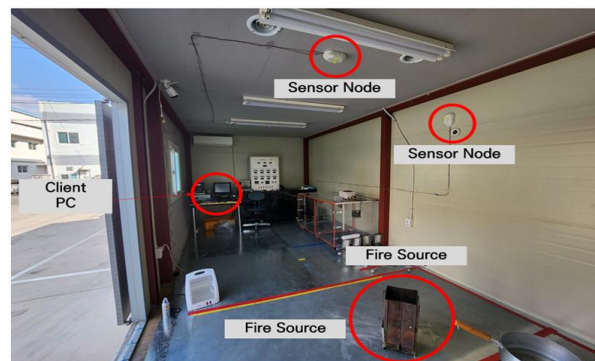


Fig. 11. The internal environment of the chamber for data collection

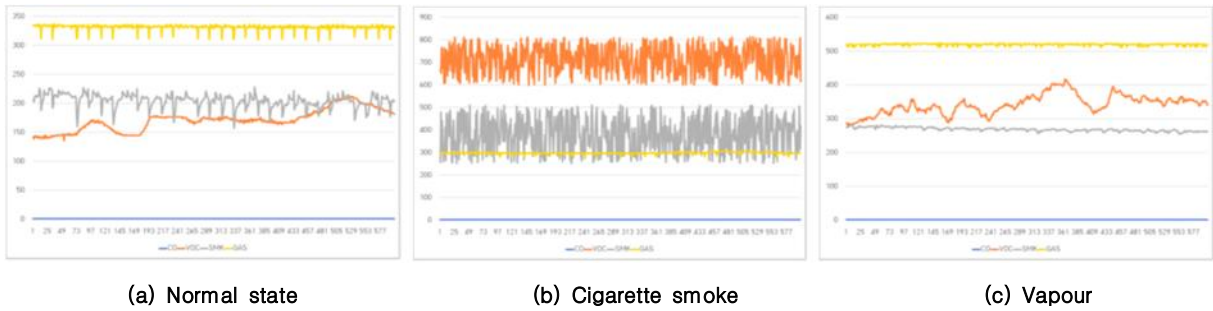


Fig. 13. Data trends for each sensor in a non-fire state

Fig. 12는 화재 상태에서 측정되는 각 센서의 데이터 변화를 보인 것이고, Fig. 13은 비화재 상태의 센서 데이터의 변화를 보인 것이다. 분류 모델 구성은 파이썬 3.9를 사용하였으며 주요 모듈은 Table 3과 같으며 주요 절차에 따른 구현 코드의 일부분은 Fig. 14와 같다.

Table 3. List of modules for classification models

Module	Specifications
pandas	- Load and preprocess data using a DataFrame
seaborn	- Visualize the distribution and relationships of data
matplotlib	- Graph creation and visualization
sklearn	- Separate training data from test data - Data normalization (preprocessing) - Assess sorting accuracy - DT / K-NN / SVM algorithms

먼저 측정된 각 데이터를 기반으로 머신러닝 알고리즘 (결정트리, K-NN, SVM)을 선택하고 온습도 데이터를 사전에 입력한다. 두 번째 단계에서는 데이터 정규화를 위하여 학습 데이터의 특징들을 기반으로 스케일링을 수행하고 Fire 레이블이 고르게 분포하도록 데이터를 샘플링한다. 또한, 학습 데이터와 테스트 데이터를 분할한다. 세 번째 단계에서는 데이터 분석 및 분포 그래프를 표현하기 위하여 Fig. 15와 같이 상관 히트맵과 분포 그래프를 출력한다. 마지막 단계에서 K-NN 분류기를 이용하여 화재와 비화재를 분류하는 모델을 완성한다. 이때 유클리드 거리 공식을 사용하여 인접한 노드의 수를 판단하여 분류를 실시하며 Fig. 14의 마지막 부분에서 화재/비화재로 분류 결과 데이터를 CLI 형태로 표현하는 것을 확인할 수 있다.

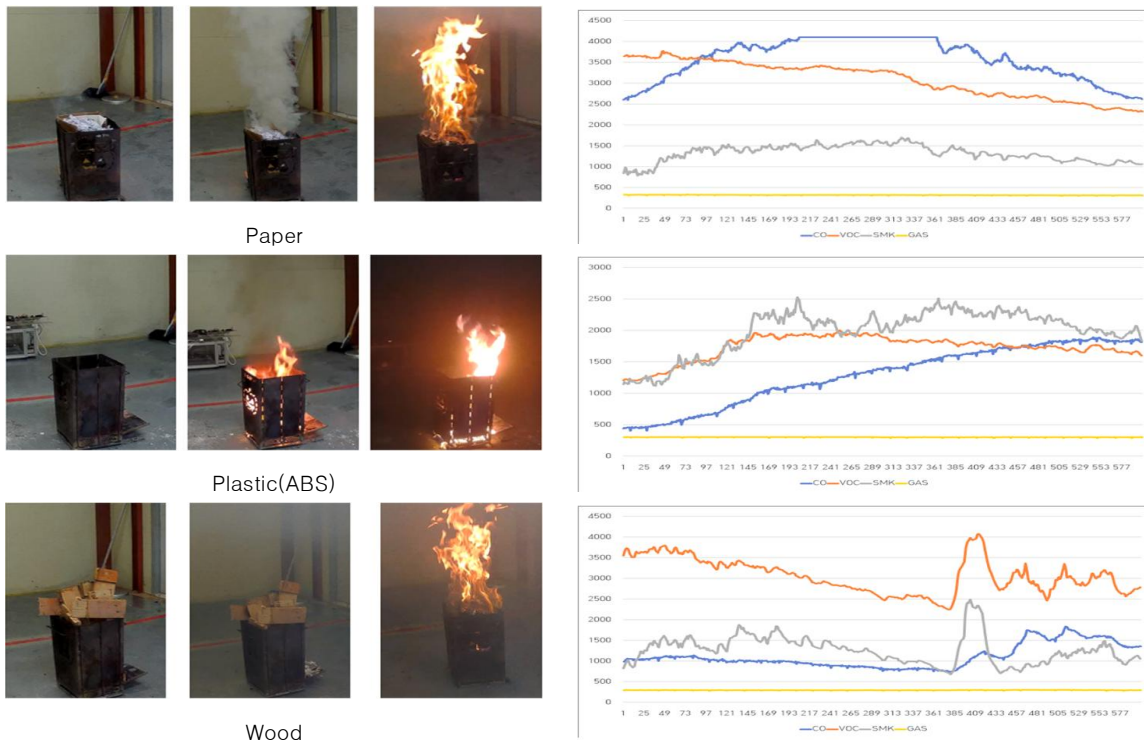


Fig. 12. Data trend of each sensor in the fire state


```

(a) learning_file = ['나무(화재).xlsx', '수증기(비화재보).xlsx', '종이(화재).xlsx', ...
feature_name = ['CO', 'VOC', 'SMK', 'GAS', 'TEMP', 'HUM']
method = (input('input ML(DT, KNN, SVM) : ').upper())
Temp_Hum_data = input('input current TEMP, HUM (ignore -> enter) : ')

(b) scalar = StandardScaler()
scalar.fit(l_data.iloc[:, :-1])
l_data = l_data.sample(frac=1, ignore_index=True)
tr_data = scalar.transform(l_data.iloc[:, :-1])
ts_data = l_data.iloc[:, -1].to_numpy()
X_train, X_test, y_train
y_test = train_test_split(tr_data, ts_data, test_size=0.3, random_state=311)

(c) sb.heatmap(data=l_data.corr(), annot=True)
sb.pairplot(l_data, hue='fire', diag_kind='kde')

(d) elif method == 'KNN':
from sklearn.neighbors import KNeighborsClassifier
ML = KNeighborsClassifier(n_neighbors=5, metric='euclidean')
ML.fit(X_train, y_train)
pred = ML.predict(X_test)
print('KNN accuracy score : ', accuracy_score(y_test, pred))

(e) input data CO VOC SMK GAS TEMP HUM (ignore -> enter) : 150 500 600 340 25 35
No Fire
input data CO VOC SMK GAS TEMP HUM (ignore -> enter) : 150 505 588 341 25 35
No Fire
    
```

Fig. 14. Part of the implementation code for the K-NN classification model

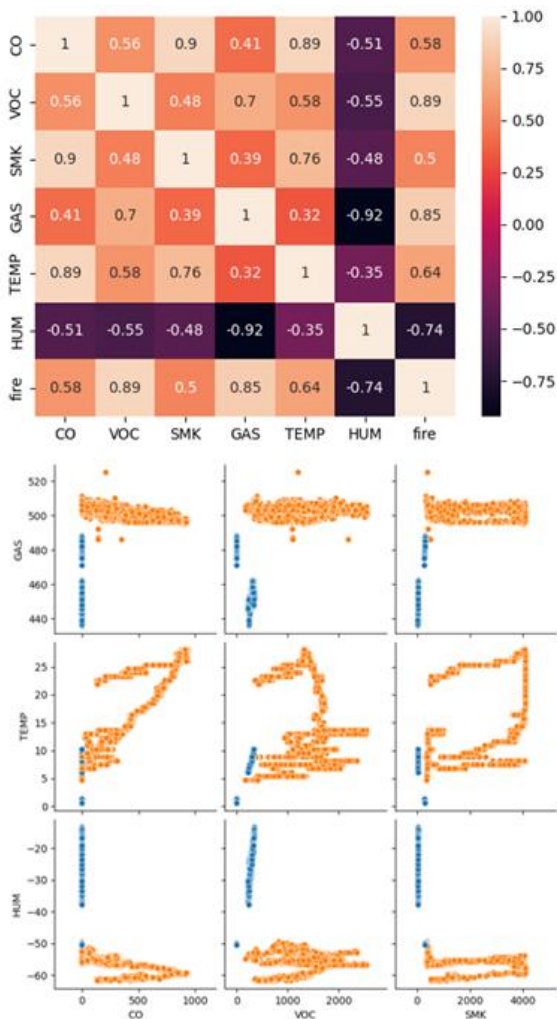


Fig. 15. Heatmaps and distribution graphs for correlation analysis

분류 모델의 성능평가가 결과는 Table 4와 같으며, 화재 분류를 위한 항목은 화재(나무, 종이, 플라스틱)와 비화재(일반상태, 수증기, 담배연기) 각 3건이다. 실제 확인 상태일 때 화재로 분류한 데이터의 비율(TPR)은 96.1%이며 비화재 상태일 때 화재로 분류한 비율(FPR)은 0%, 테스트 정확도에 해당하는 화재와 비화재 분류의 정밀도와 재현율의 조화 평균(F1-Score)은 98.01%, 분류 모델의 예측 정확도(ACC)는 98.05%로 우수한 성능을 보였다.

Table 4. Performance evaluation results of classification models

Item	Method	Result
Classification	Wood, paper, plastic, Normal, Vapour, cigarette smoke	6 cases
TPR	$(TP/(TP+FN))*100$	96.10%
FPR	$(FP/(FP+TN))*100$	0%
F1-score	$2(preci*recall)/(preci+recall)*100$	98.01%
ACC	$(TP+TN)/(TP+TN+TP+FN)*100$	98.05%

V. Conclusions

본 논문은 단일 센서 기반의 화재 감지 시스템의 정확도 문제를 해결하기 위하여 다중 센서 기반의 감지기를 제안 하였고 감지기 간의 통신을 통하여 공간 범위의 확장성과 화재 경보 정보에 대해 상호 연계할 수 있는 Wi-Fi Mesh 기반의 네트워크를 구축하여 실효성을 입증하였다. 또한, 화재 여부를 식별하기 위하여 K-NN 분류 알고리즘을 이용한 모델 구축으로 더욱 신뢰성 있는 화재 감지 시스템을 구성하였다. 다만, 분류 모델에 사용되는 학습 데이터가 화재와 비화재 유형 각 1,800건으로 한정되어 있어, 모델 적용 영역 및 다양한 분류 항목의 강화가 요구된다. 향후, Mesh 네트워크 모니터링 시스템의 강화하고 다중 센서 감지기의 자가진단 기능을 추가하여 오동작으로 인한 문제 유발을 방지하는 시스템으로 발전시키고자 한다.

REFERENCES

[1] The Kyunghyang Shinmun, "Fire trucks dispatched with alarms sound, 29 out of 30 times 'in vain'", Sep. 2023. <https://www.khan.co.kr/national/national-general/article/202309191603001>

[2] S. F. Sulthana, C. T. A. Wise, C. V. Ravikumar, R. Anbazhagan, G. Idayachandran and G. Pau, "Review Study on Recent Developments in Fire Sensing Methods," in IEEE Access, Vol.

- 11, pp. 90269-90282, Aug. 2023. DOI : 10.1109/ACCESS.2023.3306812
- [3] D. -L. Nguyen, M. D. Putro and K. -H. Jo, "Lightweight Convolutional Neural Network for Fire Classification in Surveillance System," in IEEE Access, Vol. 11, pp. 101604-101615, Aug. 2023. DOI : 10.1109/ACCESS.2023.3305455
- [4] Ryu Jinkyu, and Kwak Dongkur, "A Study on Flame and Smoke Detection Algorithm Using," Convolutional Neural Network Based on Deep Learning, J. Korean Soc. Hazard Mitig., Vol. 20, No. 1, pp.223-232, Feb. 2020. DOI : 10.9798/KOSHAM.2020.20.1.223
- [5] Han-Sol Bang, Joo-Hyung Lee, and Jin-Ku Choi, "Implementation of Fire Detection with Color Filters and Deep Learning," Journal of KIIT. Vol. 21, No. 10, pp. 137-145, Oct. 2023. DOI : 10.14801/jkiit.2023.21.10.137
- [6] Yun-ji Kim, Hyun-chong Cho, "Detecting Location of Fire in Video Stream Environment using Deep Learning," The Transactions of the Korean Institute of Electrical Engineers, Vol. 69, No. 3, p.474-479, March 2020. DOI : 10.5370/KIEE.2020.69.3.474
- [7] M. F. Ruzaj Al-Okby, T. Roddelkopf and K. Thurow, "Low-cost IoT-based Portable Sensor Node for Fire and Air Pollution Detection and Alarming," 2024 IEEE Sensors Applications Symposium (SAS), Naples, Italy, pp. 1-6, Aug. 2024. DOI : 10.1109/SAS60918.2024.10636455
- [8] Chai, Y., Zeng, XJ. & Liu, Z., "The future of wireless mesh network in next-generation communication: a perspective overview," Evolving Systems, Vol. 15, pp. 1635-1648, Aug. 2024. DOI : 10.1007/s12530-024-09583-8
- [9] ESPRESSIF, SDKs, <https://github.com/espressif/esp-mesh-lite>
- [10] M. Pujari, C. Awati and S. Kharade, "Efficient Classification with an Improved Nearest Neighbor Algorithm," 2018 Fourth ICCUBEA, Pune, India, pp. 1-5, April 2019. DOI : 10.1109/ICCUBEA.2018.8697515.

Authors



Cha-Hun Park received a B.S degree in electronic engineering from Yeungnam University in 1988, and the M.S. and Ph. D. degrees in electronics engineering from Yeungnam University in 1990 and 2001,

respectively. He is currently a Professor in Department of Avionics Engineering, Kyungwoon University. He is interested in Embedded System, SoC and Telemetry System.



Hyoun-Chul Choi received a B.S. degree in Department of Electronics Engineering from KyungWoon University in 2004. He is currently working as the CEO of Daon Co.k Ltd.

He is interested in Smart Factory, 6G AP and Security System.



Myeong-Chul Park received a B.S. degree in Computer Science from Korea National Open University in 1999, and the M.S. and Ph.D. degrees in Computer Science from GyeongSang National University in 2002 and

2007, respectively. He is currently a Professor in the Department of Avionics Engineering, KyungWoon University. He is interested in Visualization, Simulation, Education of Software, Healthcare, and DTx(Digital Therapeutics).