Scientific African 27 (2025) e02559

Contents lists available at ScienceDirect

Scientific African

journal homepage: www.elsevier.com/locate/sciaf

# Fire detection and anti-fire system to enhance food security: A concept of smart agriculture systems-based IoT and embedded systems with machine-to-machine protocol

Abdennabi Morchid [a,*], Ishaq G . Muhammad Alblushi [b], Haris M. Khalid [c,d], Rachid El Alami [a], Zafar Said [e,f], Hassan Qjidaa [a], Erdem Cuce [g,h], S.M. Muyeen [i], Mohammed Ouazzani Jamil [j]

[a] Department of Physics, Laboratory of Informatics, Signals, Automation and Cognitivism (LISAC), Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, BP 1796 Fes-Atlas 30003 Fez, Morocco
[b] Department of Electrical and Electronics Engineering, Higher Colleges of Technology, Sharjah 7947, United Arab Emirates
[c] College of Engineering and Information Technology, University of Dubai, Academic City 14143, Dubai, United Arab Emirates
[d] Department of Electrical and Electronic Engineering Science, University of Johannesburg, Aukland Park 2006, South Africa
[e] Sustainable and Renewable Energy Engineering Department, College of Engineering, University of Sharjah 27272, United Arab Emirates
[f] Department of Industrial and Mechanical Engineering, Lebanese American University (LAU), Byblos, Lebanon
[g] Low/Zero Carbon Energy Technologies Laboratory, Faculty of Engineering and Architecture, Recep Tayyip Erdogan University, Zihni Derin Campus, 53100 Rize, Turkey
[h] Department of Mechanical Engineering, Faculty of Engineering and Architecture, Recep Tayyip Erdogan University, Zihni Derin Campus, 53100 Rize, Turkey
[i] Department of Electrical Engineering, Qatar University, Doha, 2713, Qatar
[j] Systems and Sustainable Environment Laboratory (SED), Faculty of Engineering Sciences (FSI) Private University of Fez (UPF), Fez, Morocco

## ARTICLE INFO

Editor name: DR B Gyampoh

*Keywords:*
Alert e-mail
Anti-fire security system
Cloud computing
Fire detection
Food security
Internet of things (IoT)
Message queuing telemetry transport (MQTT)
Simple mail transfer protocol (SMTP)
Smart agriculture
Sustainability

## ABSTRACT

Food security has become a major concern for most countries. This is due to: 1) the growth of the world population, 2) the decline of natural resources, 3) the loss of agricultural land, and 4) the increase of unforeseen environmental conditions (storms, fires, and other natural hazards). The fire outbreak, in general, has developed into a serious concern. In the coming years, the rate of fire outbreaks could rise exponentially, requiring immediate attention to avoid loss of property and life. To resolve such an issue, a shift from the agricultural industry to smart agriculture via applications of 1) the Internet of Things (IoT), 2) embedded systems, and 3) sensors for fire prevention are required to improve operational efficiency and productivity. A fire detection and anti-fire security (FDAS) system in smart agriculture using the IoT and embedded system is proposed. The proposed system has four technology levels: 1) the edge network layer, 2) the fog network layer, 3) the cloud computing layer, and 4) the data representation layer. The proposed system uses an embedded system like a Raspberry Pi device and sensors to measure the amount of fire smoke in the air and the proportion of fire in the area. The data obtained from the sensors are sent over the internet to the ThingSpeak platform using the machine-to-machine-based Message Queuing Telemetry Transport (MQTT) protocol for further display and analysis. Data available is then 1) stored, 2) processed, and 3) visualized through the ThingSpeak platform in real-time. An e-mail alert is sent to the farm owner if a fire is detected on the farm using the Simple Mail Transfer Protocol (SMTP). When a fire is detected, the anti-fire system is activated. It further

---

* Corresponding author.
*E-mail address:* abdennabi.morchid@usmba.ac.ma (A. Morchid).

https://doi.org/10.1016/j.sciaf.2025.e02559
Received 27 June 2024; Received in revised form 27 October 2024; Accepted 16 January 2025

filters and analyzes the data using the MATLAB application. Python programming language is also used to develop the program source code. The performance results of the proposed scheme show an accurate fire detection and anti-fire system performance for improving food security and sustainability in agriculture.

## Nomenclature

*Acronyms*

| | |
|---|---|
| Adruino IDE | Adruino integrated development environment |
| ADC | Analog-to-digital converter |
| AI | Artificial Intelligence |
| EDA | electronic design automation |
| FAO | Food and Agriculture Organization |
| FDAS | Fire Detection and Anti-Fire Security |
| GPIO | general purpose input/output |
| GRID | global resource information database |
| IIoT | Industrial Internet of Things |
| IoT | Internet of Things |
| LAN | local area network |
| LINUX | lovable intellect not using XP |
| LSB | least significant bit |
| MATLAB | matrix laboratory |
| MCP | microchip converter programmable |
| MQTT | message queuing telemetry transport |
| MTAs | message transfer agents |
| OS | operating system |
| PC | personal computer |
| PCS | personal communications service |
| PoE | power over ethernet |
| SMTP | simple mail transfer protocol |
| SMTPS | simple mail transfer protocol secure |
| SPI | serial peripheral interface |
| SSL | secure sockets layer |
| TLS | Transport Layer Security |
| UNEP | united nations environment program |
| UAV | Unmanned Aerial Vehicle |
| USA | United States of America |
| VNC | virtual network computing |
| Wi-Fi | wireless fidelity |

*Symbols*

| | |
|---|---|
| $V_{out}$ | digital output voltage |
| $V_{in}$ | analogue input voltage |
| $V_{ref}$ | reference voltage |
| $Y_{cal}$ | calibrated output |
| $V_{ncal}$ | non-calibrated output |

## Introduction

### *Agriculture – rise in demand, resources, and world population*

Agriculture is one of the most important economic sectors in the world. It provides the food needs of nearly 7.5 billion people. This food requirement is increasing at a breakneck pace, which could increase the demand for food and animal materials [1]. For example, the United States of America (USA) produces large quantities of meat by occupying second place in the world after Brazil. These large quantities are in the production of meat for preservation which puts tremendous pressure on agricultural resources by raising livestock, etc. This pressure exerted by humans on agricultural and forest resources constitutes a real threat to humans, as recognized by the Food and Agriculture Organization (FAO) [2]. According to estimates, the world population will reach 10 billion by 2050. According to the United Nations FAO, food production will need to expand by around 70 % to feed this growing population [3,4].

### *Serious threat to food security, concept of smart agriculture, and motivation of this work*

According to a new analysis by UN Environment Program (UNEP) and the global resource information database (GRID)-Arendal, climate change and land-use change are expected to increase the frequency and intensity of wildfires. These wildfires will have a global rise of 1) up to 14 % by 2030, 2) 30 % by the end of 2050, and 3) 50 % by the end of the century [5]. This could affect meat production and can dramatically affect the vegetation too. This is due to the vast areas of forests that are uprooted to secure natural pastures. The same is the case with the Amazon Forest in Brazil [6,7]. In addition, fires pose a serious threat to food security in the world. Especially in recent years due to climatic changes that have led to a rise in global temperature, as is the case with the fire that killed large areas of the Amazon Forest in 2020 [8,9].

Similarly, North Africa was not spared from the scourge of fires in this pursuit. It has recently been known that fires that destroyed agricultural products in northern Morocco significantly impacted the farmers who were affected by the lack of crops [10–12]. In the face of this dangerous pest, whose severity is increasing dramatically due to several human and natural factors, there has become an urgent need to find effective scientific alternatives to avoid these fires before the fire intensify to protect the population's food security, which is the primary motivation of this work. The aim is to fill the gap that has been developed in the agriculture field through time due to the danger caused by fire outbreaks. Based on this, smart agriculture which relies heavily on technological development plays a significant role in increasing agricultural production. This also plays a major role to meet the world population's needs while minimizing and avoiding fires that could pose a substantial threat to global food security [13–15].

### *Intelligent systems, forest monitoring, and control*

Intelligent systems were created at a technological level to answer environmental and social concerns. These systems have made the lives of people more profitable and productive worldwide. A wide range of applications is being developed for these goals. Smart forests look to be a viable option for natural ecological long-term viability with features such as: 1) environmental health assessments, 2) climate change, 3) animal monitoring, 4) natural disaster early warning, and 5) monitoring systems [16]. Forests encompass about four billion hectares or about one-third of the world's land surface. Plants and animals dominate it. Forests have a tremendous environmental influenced by: 1) preventing warming to a greater extent, and 2) reducing pollutants created by humans. However, the forest fire could pose a significant hazard. It has the potential to: 1) unleash an enormous environmental disaster, 2) increase human and wildlife deaths, 3) hurt the ecosystem and economy, and 4) biodiversity. Forest fires spread swiftly and go unchecked in many nations harming and killing many people and animals yearly [17]. This makes forest monitoring and controls a significant issue. Adequate solutions must be found to reduce forest fires while protecting vital resources.

### *New technology and smart agriculture*

New technologies such as the IoT are increasingly deployed in agriculture systems. Most farmers use large tracts of farmland making every turn challenging to reach and follow. As a result, IoT technology helps maintain and expand agricultural productivity. [18–23]. This article discusses the difficulty of continuously transmitting and processing large quantities of Big Data in industrial environments. Additional data security measures are required due to the increasing use of cloud platforms to store huge amounts of data on unreliable servers. However, existing authentication mechanisms for Industrial Internet of Things (IIoT) devices suffer from drawbacks, including single-factor authentication and low adaptability, compounded by the growing number of IIoT devices and the diversity of end users [19]. On the other hand, this study [20] examines the integration of technology in agriculture. It focuses strongly on the use of smart devices to monitor crops in real-time. The study addresses privacy and security issues in the context of smart agriculture, with a focus on precision farming and the possible effects of cyber attacks on economies that rely heavily on agriculture. Focusing on a review of associated devices and communication methods, this study examines the benefits and challenges of integrating wireless sensors and IoT in agriculture [21]. An IoT-based intelligent agricultural system that streamlines farming procedures is presented in the study [22]. Farmers benefit from improved production efficiency and product quality thanks to the integration of intellectualized products, which also reduces time-related costs and increases crop yields. In addition, an IoT-based sound detection algorithm was presented by the authors of the study [23] to detect red palm weevil larvae in date palms. This technology is a viable option for early pest control in smart agriculture due to its high accuracy in identifying infestations.Fires are considered one of the unexpected events that have an impact on farm production. The fires have developed into a serious concern requiring immediate attention to avoid loss of property and life. This has put the lives of farmers, agricultural production, and farmed animals at grave risk due to the fire outbreak [24]. The study's authors [25] focus on the real-time detection and prediction of abnormal conditions using a wireless equipment package for fire water systems. The main objective of this study is to improve fire suppression efficiency. Rescue and firefighting efforts are critical, and hidden or dry hydrants can lead to firefighting failures. This study [26] implements an intelligent fire detection system using sensors such as smoke and flame sensors that simultaneously protect people and valuable property by alerting homeowners, emergency services, and police stations in the fire district. In this work [27], a dataset of real forest fires was used to identify the real-time behavior of IoT sensors. This will facilitate secure operational efficiency, throughput, and field network management decisions for the precision agriculture sector. To avert these losses, one way is to respond immediately to this situation. Early fire detection followed by an anti-fire system will protect the farm from loss of life and property by installing a 1) fire alarm system, and 2) automatically activating sprinklers to put out the fire proposed in this study [28–34]. This article [29] presents a wireless IoT system for monitoring indoor air quality. It uses air quality, temperature, and humidity sensors and a Raspberry Pi 2. The system provides real-time data, stores it in the cloud, and sends e-mail alerts in the event of high pollution levels. It offers an affordable, portable solution for improving indoor air quality, with a direct impact on the health of occupants. In addition, this article [30] highlights the loss of 11.9 million hectares in forest fires in 2019 and the continuing increase in the area under threat. Recent fires in Australia have resulted in the death of a billion animals, including 25,000 koalas. To prevent these disasters, it is crucial to detect fires early. To this end, we are using an ESP32 card with rain, sound, DHT11, and PIR sensors. Further, This article [31] proposes the design and implementation of a fire safety system for controlling and monitoring a siren using a new on- board, in order to autonomously manage fire detection and extinguishing in a Smart Farm. This innovative system, called Fire Safety System Smart Farm (F3SF), is based on the Atmega328p microcontroller and enables continuous monitoring of the Smart Farm in the event of a fire. Thanks to specific sensors connected to the F3SF board, the system automatically detects fires and controls and automatically maintains the sirens and sprinklers used to extinguish them. The F3SF-0X version of the system is implemented using a voltage sensor, an H-bridge, an ATMega328p microcontroller, a flame sensor, an MQ-135 gas sensor, a water sprinkler, and an electronic relay. The system is simulated using the EasyEDA web-based software tool. This paper [32] presents the development of an embedded system for intelligent lighting and ventilation, designed to detect human presence in order to control lighting and monitor parameters such as humidity, temperature, $CO_2$ and smoke in real-time, thus ensuring optimal ventilation and the prevention of fire and smoke-related accidents. The system incorporates ready-to-deploy software, enabling real-time monitoring via HTTP protocol, accessible from a smartphone or computer. In addition, sensor data is stored in real-time on a cloud server, enabling users to monitor this information remotely from anywhere in the world. On the other hand, this paper [33] presents an intelligent technology for continuously assessing the fire scene and associated risks by analysing variations in the sound field. It focuses on quantifying the attenuation of sound pressure caused by fire, thus providing a sound scientific basis for the development of this technology. The fire monitoring system is demonstrated through the use of a real alarm and low-intensity acoustic sensors, illustrating its effectiveness in real-life conditions. In addition, a roadmap is proposed for the application of this alarm-based monitoring method in intelligent firefighting. Although many systems exist and are highly developed in actual circumstances, the proposed ones have been merely deployed. Moreover, the proposed systems are 1) unstable in automation, 2) expensive, and 3) difficult to install.

### *Agricultural practices, artificial intelligence, and focus of this work*

Agricultural practices are changing dramatically as image processing and analysis tools are combined to give a simple solution and conquer the difficulties of agricultural fires. In this regard, this paper [35] proposes a system for early fire detection using sensor networks and Unmanned Aerial Vehicle (UAV) technologies. The system analyzes environmental factors by combining cloud computing, UAVs, and wireless sensors. In addition, image processing methods improve the accuracy of fire recognition. Artificial Intelligence (AI) is a new technique and method used in the agricultural field for fire detection. Further, this paper [36] discusses the technological and economic benefits of AI applications for agricultural infrastructure safety. Forest fires cause significant losses to the agricultural sector, including damage to livestock buildings and crops, as well as disruption to day-to-day operations. While these strategies are beneficial, they have little practical use on the farm. The case for AI is based on accurate computer simulations used for risk assessment, smoke movement analysis, and post-incident analysis. Fire combustion produces: 1) smoke, 2) heat, 3) flame, 4) gas, 5) temperature, 6) smoke concentration, 7) carbon monoxide, and 8) by-products. Also, temperature, smoke concentration, and carbon monoxide are all characteristics of fires. For this purpose, it used sensors to detect fires [37]. This article [38] presents the design and development of a multi-sensor fire detection system based on fuzzy logic, as well as a notification system via a web platform. Until recently, the majority of consumer fire detection systems were based exclusively on smoke detectors. It has been shown that the protection offered by these devices is limited by the type of fire and the detection technology used. This problem is exacerbated by the lack of effective warning and notification mechanisms. Indeed, a conventional system often requires the physical presence of a person to respond to the alarm. In developing countries, poor planning and management have a negative impact on the response times of fire brigades and rescue teams. To overcome these shortcomings, a fuzzy logic system was implemented using an Arduino development board, integrating data from an MQ2 smoke sensor, a temperature sensor and a flame sensor. Information from the detection system is transmitted in real time by SMS (Short Message Service), via a GSM (Global System for Mobile Communication) SIM900 module, to the web system and the homeowner or manager. Access to the web system also enables fire brigades and rescue teams to receive real-time information on the location of the incident. In addition, with the world's population growing rapidly, traditional farming methods are no longer sufficient. It is therefore crucial to adopt innovative agricultural techniques to meet growing food needs. This paper [39] presents smart agricultural systems integrating embedded technologies and IoT, as well as wireless sensor networks for agriculture and livestock. It describes systems, electronic circuits, communication protocols, and intelligent monitoring solutions accessible via PCs and smartphones, while proposing recommendations and discussing future prospects for technologies in intelligent agriculture. Furthemor, this study [40], presents the design and implementation of a fire safety system designed to control and monitor a siren using an Arduino Uno. The aim of this system is to reduce the risks during the evacuation process, in order to protect both medical staff and patients at Morocco's University Hospitals. To tackle the problem of managing and monitoring the siren in the event of a fire, we developed a new device capable of controlling the siren during an incident and transmitting its status to the control and signalling equipment. The hardware used was an Arduino UNO, a voltage sensor, an H-bridge, a voltage regulator, a bipolar transistor, an optocoupler, an LCD screen and the siren. On the software side, the Arduino IDE and EasyEDA were used. On the other hand, the paper [41] mainly presents the development of a wireless sensor network dedicated to intrusion and fire detection. A sensor network model has been designed to monitor temperature and carbon monoxide levels. A device is proposed as a wireless sensor unit to perform this monitoring. In addition, a method is put forward for identifying variations in temperature and carbon monoxide levels, enabling intrusion and fire detection. Finally, the paper investigates the data transmission process within a wireless sensor network to optimise fire and intrusion detection. The ThingSpeak platform is used which saves data from sensors in real-time. The farm owner can further download and analyze this data obtained over the internet to the ThingSpeak platform via MQTT Protocol [42,43]. Simple Mail Transfer Protocol (SMTP) is also used because it is a highly secure and efficient way to transmit notifications to e-mail [44]. Also, the sensors generally used for fire detection and anti-fire systems to enhance food security similar to those used in many applications are subject to measurement noise and stochastic disturbance where measurement noise is zero mean white noise process while stochastic disturbances are the output of a linear system driven zero mean white noise. In such cases, noise echo cancellation is recommended for restoring the sensor performance [45]. To improve farming practices, smart sensing in sustainable agriculture involves the use of technologies such as cloud platforms, deep learning, and machine learning [46]. This article [47] presents concrete examples of promising materials for the production of wireless devices for the transmission and analysis of digital information. In addition, this article [48] provides concrete examples of modelling and solving optimisation problems. This work focuses on detecting and preventing fire outbreaks in real time by utilizing 1) IoT, 2) embedded systems, and 3) communication protocols.

### *Main contribution of this work*

The main contribution of this work is the proposed Fire Detection and Anti-Fire Security (FDAS) system. The system is proposed in a smart agriculture environment using the IoT and embedded systems while improving operational efficiency and productivity. The graphical abstract of the proposed scheme can be seen in Fig. 1. The proposed system has four layered identifications: 1) the edge network layer (at its most basic level, edge computing brings computation and data storage closer to the devices where it is being gathered rather than relying on a central location that can be thousands of miles away), 2) the fog network layer (a fog device (or fog node) is a highly virtualized IoT node that provides computing, storage, and network services between edge devices and the cloud. It has characteristics like those of the cloud), 3) the cloud computing layer (it is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power without direct active management by the user), and 4) the data presentation layer (the proposed system uses an embedded system like a Raspberry Pi is small single board computer). The proposed system uses an embedded system like a Raspberry Pi device and sensors such as a fire smoke sensor and flame sensor to measure the amount of fire smoke and the amount of smoke in the air. The data obtained from the sensors are sent through the internet to the ThingSpeak platform via the MQTT protocol for further visualization and analysis. Data available is 1) stored, 2) processed, and 3) visualized through the ThingSpeak platform in real-time. Once the data is analyzed, an e-mail notification on fire detection is sent to the farm owner. The owner, in return, confirms to alert the firefighters and an activated anti-fire system. When a fire is detected, the proposed firefighting system aims to operate at a record speed and with an average e-mail sending time of 1.75 s. This would also allow

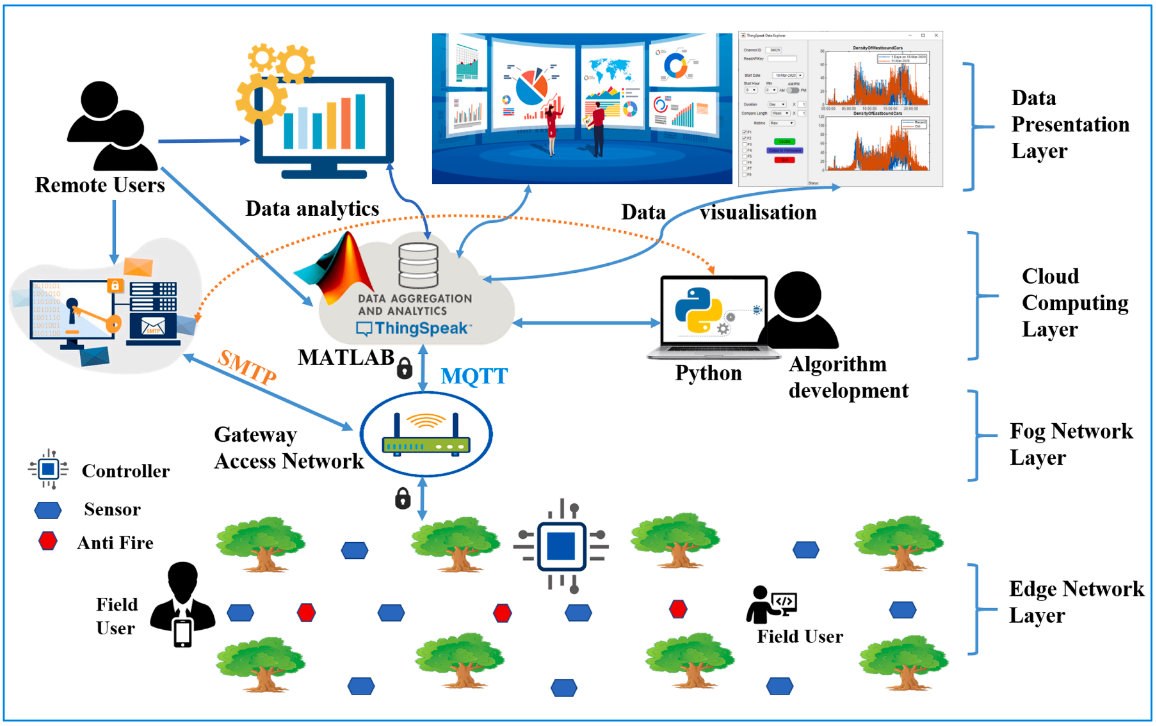


**Fig. 1.** Graphical abstract of the proposed FDAS system[1]

[1] SMTP and MQTT are the acronyms of simple mail transfer protocol and message queue telemetry transport, respectively.

the farmer to monitor his farmed crops while sitting anywhere remotely and with a minimal cost. Note that the MATLAB application prepares, filters, and analyzes the data. The Python programming language is utilized to develop the program source code. The performance results of the proposed scheme show an accurate fire detection and anti-fire system performance for improving food security and sustainability in agriculture.

### *Formation and layout of this paper*

The formation of the rest of the paper is as follows. The research methods are described in Section 2. In Section 3, a proposed algorithm of the FDAS system is represented. Results and discussion of the proposed FDAS system are presented in Section 4. The conclusions and future work are presented in Section 5. The layout of this work can be seen in Fig. 2.

## Research methods

The Internet of Things (IoT) technology has many applications. This technology and its use are rapidly increasing in smart agriculture. It functions according to how it was designed and developed. However, there is no globally accepted standard-defined architecture for functioning. The architecture of the IoT is determined by its usefulness and application in various domains. This section will discuss the proposed IoT architecture and block diagram of the FDAS system.

### *IoT architecture of the FDAS system*

Fig. 1 shows the architecture of the proposed system that has been set up to collect and process data in agriculture effectively. This system consists of four layers: 1) the edge network layer, 2) the fog network layer, 3) the cloud computing layer, and 4) the data display layer. The proposed IoT architectural layer functionality for the FDAS system is addressed as follows: Firstly, the edge network layer comprises physical network sensors used in the system. These sensors are fire smoke sensors, flame sensors, and anti-fire systems. The functionality of these sensors is to block fire. The farm owner or laborers can interact with the system immediately. Secondly, the fog network layer contains only the router (access network) as an intermediate to connect the edge and cloud computing layers. Thirdly,

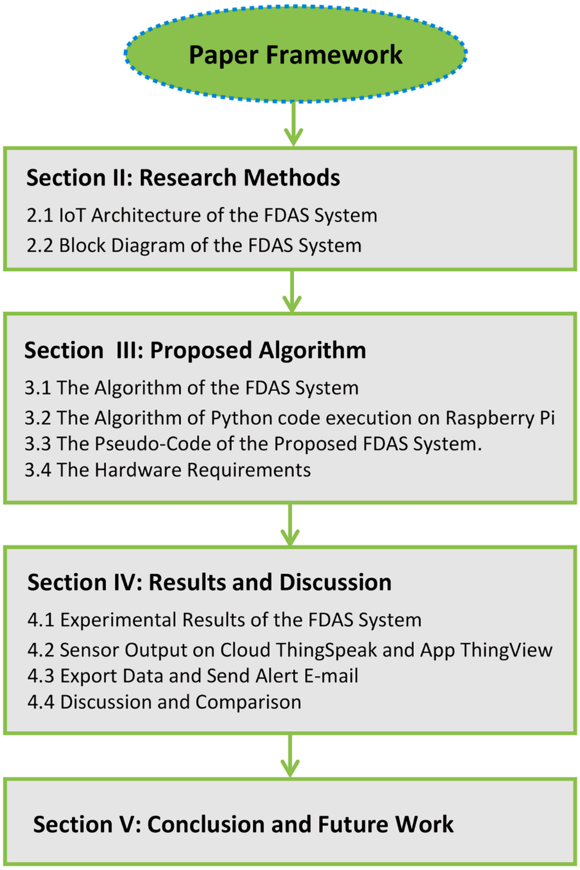


**Fig. 2.** Framework of this paper.

the cloud computing layer is the heart of the proposed system. It contains the algorithm developed to control the system: the cloud Thingspeak platform and the SMTP server. Fourthly, the data presentation layer in this last layer lets the users and administrator of the system handle all the data acquisition on the MATLAB platform which are: 1) visualization, 2) interpretation, 3) analyzing, and 3) controlling the system. A low-cost sensor node may monitor various environmental parameters such as 1) dampness, 2) fire smoke, and 3) flame. MATLAB code in the ThingSpeak cloud application sends this data through IoT to mobile phones or PCs. ThingSpeak is a cloud-based IoT analytics platform for farmers that allows them to 1) collect, 2) view, and 3) analyze real-time data streams. A farmer may use these gadgets to feed data into ThingSpeak allowing them to create real-time visualizations of the data. ThingSpeak is an IoT analytics program that assists farmers by: 1) collecting, 2) displaying, and 3) analyzing real-time data streams before acting on the information. The farmer may study the data and submit it to the ThingSpeak Cloud or ThingView.

### *Block diagram of the FDAS system*

The block diagram of the proposed FDAS, shown in Fig. 3, consists of several essential levels. The system is divided into four main functional blocks: 1) sensors, 2) control unit, 3) actuators, and 4) display of collected data. Firstly, the sensor block includes a smoke sensor, which is responsible for detecting smoke particles in the air, and a flame sensor, which completes the detection by identifying the direct presence of flames. These sensors transmit analog signals to the MCP3208 analog-to-digital converter (ADC), a component with 8 channels and 12-bit resolution, chosen for its affordability and ease of integration. As the Raspberry Pi 3 has no analog inputs, the MCP3208 is essential for converting analog signals into digital signals that the microcontroller can process.

The control unit, consisting mainly of the Raspberry Pi 3, manages the conversion of digital signals and processes sensor data in real-time. Thanks to the ThingSpeak platform integrated into the cloud block, the data can be viewed and analyzed in real-time. Data is transmitted to the cloud platform using the MQTT protocol, providing a stable and efficient IoT connection for remote monitoring. To maximize accessibility, the ThingView application is also implemented, making it easy to consult the data on mobile devices and personal computers. SMTP protocol is integrated for the transmission of e-mail alerts, sending notifications directly to farm owners and fire brigades if a potential fire is detected. This communication system ensures a rapid response by instantly alerting the relevant parties to a critical situation. Finally, an actuator system is provided, controlling fire-fighting pumps that can be activated automatically if a fire is detected, enhancing the safety of the farm site. So, the block diagram in Fig. 3 illustrates a complete, real-time fire monitoring system, combining detection, control, communication, and rapid response to provide optimum protection for farms. Since the Raspberry Pi lacks analog inputs, many available sensors are inconvenient. It was required to expand the capabilities of the FDAS system by adding extra sensors. Therefore, a simple and inexpensive solution was considered. The solution was the MCP3208. The MCP3208 is an 8-channel analog-to-digital converter with a 12-bit resolution (ADC) [50]. It is inexpensive, simple to attach, and requires no other components. It uses the SPI bus protocol, supported by the GPIO header on the Pi.

The application of MQTT and SMTP in intelligent agriculture, particularly for fire detection, improves the efficiency, reliability and scalability of communication systems. It enables real-time monitoring, rapid response and automation potential to mitigate the impact

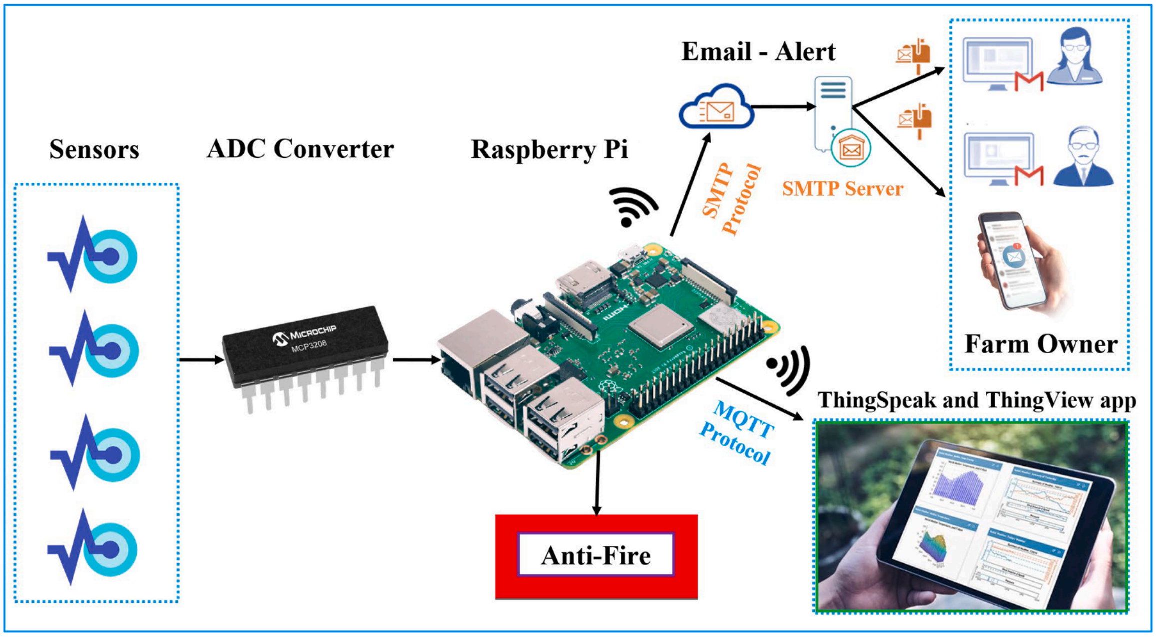


**Fig. 3.** Block diagram of the FDAS system[2].

[2] ADC, SMTP, and MQTT are the acronyms of an analog-to-digital converter, simple mail transfer protocol, and message queue telemetry transport, respectively.

of fires in agriculture. The detail of SMTP and MQTT as follows:

- MQTT, short for Message Queuing Telemetry Transport, is a standardized messaging protocol designed to facilitate machine-to-machine communication, particularly in the, field of the IoT. It offers a prominent solution for efficient data transmission between IoT devices, thanks to its smooth implementation, small code footprint, and ability to ensure seamless connectivity with millions of devices. MQTT guarantees reliability even in unreliable network environments, while facilitating message encryption and device authentication, ensuring secure communication. With extensive support for several programming languages, including Python, MQTT is a fast and efficient solution for a variety of application scenarios.
- SMTP stands for Simple Mail Transfer Protocol. It is a communications protocol used to send and receive e-mail messages over the Internet. Mail servers and other Message Transfer Agents (MTAs) use SMTP to send, receive, and relay e-mail messages. Simple Mail Transfer Protocol Secure (SMTPS) enhances the security of SMTP by using the transport layer, ensuring authentication of communication partners, data confidentiality, and integrity. This protocol relies on Secure Sockets Layer (SSL) or (Transport Layer Security (TLS) to establish a secure connection, thus preserving the confidentiality and integrity of e-mail exchanges. Although the client and server continue to use the standard SMTP protocol at the application level, the connection is secured by SSL or TLS. In SMTP architecture, the sender's client or mail server plays the role of SMTP client, establishing a secure connection with the server and transmitting the e-mail with recipient details, subject, and message content. The recipient's server processes the e-mail and determines the next appropriate server based on the recipient's address, which may be another SMTP server on the transmission path or the recipient's final mail server.

After presenting the proposed cloud-based IoT architecture for agriculture and the block diagram of the proposed system, the algorithms of the FDAS system and python code execution on Raspberry Pi are expressed.

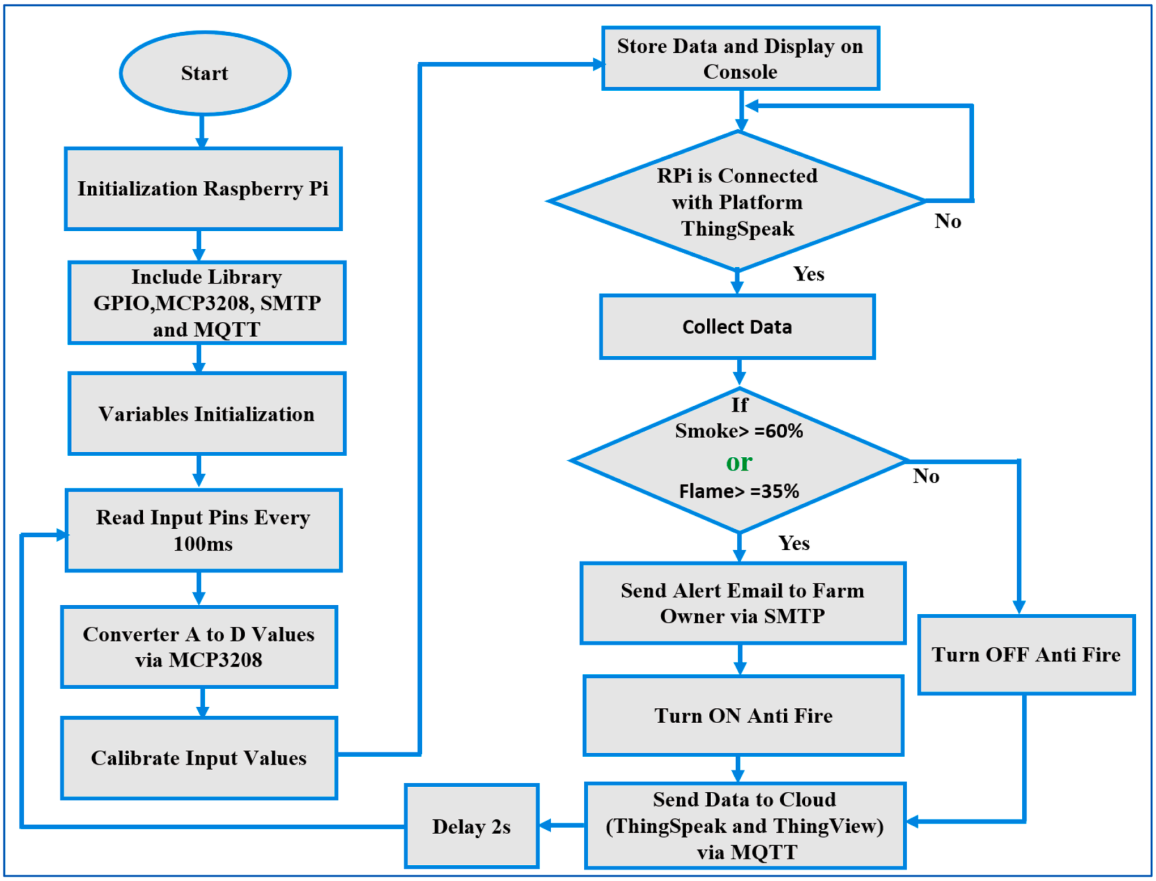


**Fig. 4.** The algorithm of the FDAS system[3].

[3] GPIO, MCP, SMTP, MQTT, and RPi are the acronyms of general-purpose input/output, microchip converter programmable, simple mail transfer protocol and message queue telemetry transport, and Raspberry Pi, respectively.

## Proposed algorithm of the FDAS system

Due to population expansion, the proposed FDAS system using IoT technology offers many advantages, such as 1) smart -efficient management of resources, 2) knowledge development, 3) improved agricultural sustainability, and 4) accurate real-time monitoring to avoid losing crops. This section will discuss 1) the flow chart for an FDAS system, 2) the flowchart of python code execution on Raspberry Pi, and 3) the pseudo-code of the proposed FDAS system.

### *The algorithm of the FDAS system*

The FDAS system algorithm is shown in Fig. 4. It comprises three columns. First, the Raspberry Pi was initialized. Some necessary libraries, like the GPIOZero library, are included. This library has interfaces for various commonplace components and more complicated items like sensors and analog-to-digital converters. This involves the following steps: 1) configure the fire smoke and flame sensor variable, 2) read data from sensors every 100 ms, 3) convert data from analog sensors into a digital signal via MCP3208 converter, 4) adjust values of the sensors such as fire smoke sensor and flame sensor, 5) calibrate data to visible values in percentage. 6) The MCP308 library must be included for converting analog sensor values to digital, 7) the SMTP library for sending alert e-mails, and 8) the MQTT library for sending data to the cloud. 9) The data is then further stored, and 10) displayed on the Python console in Raspberry.

Also, the Raspberry connection to the ThingSpeak platform is required to be checked. If there is no connection with the platform ThingSpeak, re-checking the connection is needed until the connection is made. If the connection is made, the collected data obtained from the sensors is passed. Regarding the software core for the proposed FDAS system, if the fire smoke sensor value is greater than 60 % or the flame sensor value is more than 35 %, it means there is a red fire alert on the farm. An e-mail is urgently sent to the farm owner with an alert that his farm is on fire to contact the firefighters immediately and that our proposed system has started activating the firefighting protocol. Likewise, all the data obtained through the IoT service is sent to the ThingSpeak platform and to a mobile application, ThingView, via the MQTT protocol. Here it is possible to log in to both the platform and the application and get all the data

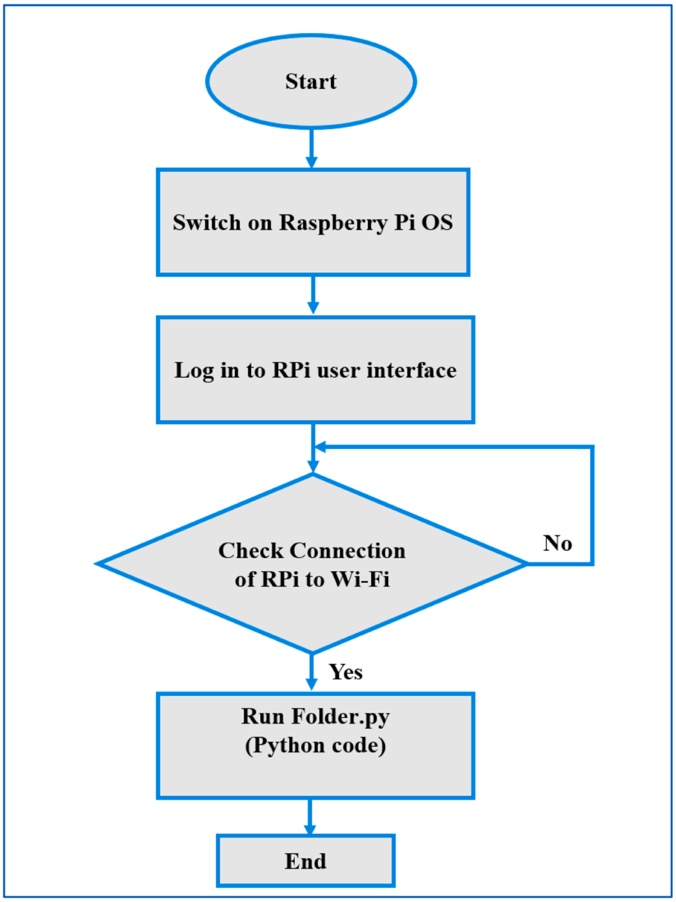


**Fig. 5.** Flowchart of Python code-based algorithm execution on Raspberry Pi[4].

[4] OS and Wi-Fi are the acronyms of operating system and wireless fidelity, respectively.

related to the proposed system. This data can also be downloaded and sent back to the cloud every 2 s for the proposed FDAS system. In this proposed algorithm, the choice of thresholds for smoke ≥ 60 % and flame ≥ 35 % is the result of extensive testing and calibration. These values were determined after a series of field and laboratory tests to assess the accuracy and sensitivity of the sensors, ensuring reliable fire detection while minimizing false positives. The 60 % smoke threshold has been set to detect a concentration that reliably indicates a potential fire risk, without being overly sensitive to non-fire-related particles. Similarly, the 35 % threshold for flame detection enables significant and continuous heat signatures to be detected, differentiating genuine fire situations from minor heat sources. These calibrated thresholds ensure that the system responds quickly and efficiently while maintaining the right balance between accuracy and reliability in the agricultural context.

### *Flow chart of python code algorithm execution on Raspberry Pi*

The flowchart of the Python code execution algorithm on Raspberry Pi is depicted in Fig. 5. After turning on the Raspberry Pi, it was checked that it is connected to the Wi-Fi network and logged in with the username and password to the Raspberry Pi user interface via the virtual network computing VNC Viewer. After switching on the Raspberry Pi, a wait is required for the connection check of RPi to Wi-Fi to be established. If the connection is not established, the data is sent to the cloud of ThingSpeak. All data sent by sensors attached to Raspberry Pi are available, displayed on the application by ThingView, and sent an alert e-mail via SMTP protocol. One of the most excellent aspects of working with Python on the Raspberry Pi is that it is treated as a priority syntax. The Raspberry Pi

**Table 1**
Pseudo code of the proposed FDAS scheme.

Algorithm. *Fire detection and anti-fire security system algorithm representation.*

```
Define:
    Fire smoke sensor – FSs
    Flame sensor – Fs
    Seconds – s
    Initialize RPi – IRPi
    Initialization of fire smoke sensor – IFSs
    Initialization of flame sensos – IFs
    Update fire smoke sensor – UFSs
    Update flame sensor – UFS
    Digitalization of fire smoke sensor – DFSs
    Digitalization of flame sensor – DFSs
    Calibration of fire smoke sensor – CFSs
    Calibration of flame sensor – CFs
    Storage and display on the console – SDc
    Collecting data – CD
    App ThingView – ATv
    Cloud ThingSpeak – CTs
    Turn on anti-fire – TonAF
    Turn off anti-fire – ToffAF
    Send alert e-mail – SAE
    Cloud ThingSpeak – CTs
    App ThingView – ATv
    Send data to cloud ThingSpeak – SDCTs
    Send data to app ThingView – SDATv
    Inputs: {FSs:Fs} ← Sensors
    Output: {TonAF: ToffAF: SAE: SDCTs: SDATv} ← State of the FDAS System
    Initialize RPi
    Include Libraries: {GPIO: MCP3208: SMTP: MQTT}
    Initialization of FSs, Fs
    Begin
    Repeat for each period of 2S:
            UFSs, UFs ← update (FSs, Fs)
            DFSs, DFs ← digitalization (FSs, Fs)
            CFSs, CFs ← calibration of the (FSs, Fs)
            SDc of FSs, Fs ← Storage and display on the console of the (FSs, Fs)
        if (RPi is connected with CTs)
            CD
            if (FSs ≥ 60 % or Fs ≥ 35 %)
                SAE
                TonAF
            else
                ToffAF
            SDCTs, SDATv ← SD to CTs and ATv
            end if
        else
    Trying to connect RPi to CTs
    End Repeat
```

foundation chose Python as the primary language because of its 1) strength, 2) adaptability, and 3) ease of use. Python is pre-installed on the Raspberry Pi so that the start can be obtained immediately. Many programmers still prefer Python because it is much easier to use. Python hides much information from the programmer, which can aid with debugging. The type of each variable is not required to be described in the code since Python is a dynamically typed language; Python will do it for us. In statically typed languages (such as C, C++, or Java), on the other hand, the kinds of variables must be defined.

### *The pseudo code of the proposed FDAS system*

The pseudo-code is presented here to program, develop, and plot the algorithm's structure. Table 1 provides the terminology for all operations used in pseudo-code, including input, processes, and output from line # 1 to 24. It also represents the algorithm of the proposed FDAS system. Here lines # 25 and 26 illustrate the input and output of the system. Lines # 27 to 29 represent the initialization of RPi, including the libraries and initialization of FSs and Fs. Lines # 31–47 include the repeat function. This consists of the 1) function that this system will operate with, 2) starts from reading the sensors, 3) processing these values, 4) then giving commands to the outputs, and 5) includes extinguishing the fire and sending data to the cloud. The repeat function contains: 1) update of the FSs and Fs (UFSs, UFs); 2) digitalization of the FSs and Fs (DFSs, DFs); 3) calibration of the FSs and Fs (CFSs, CFs); 4) storage and display on the console (SDc) of the FSs and Fs;5) trying to connect RPi to cloud ThingSpeak (CTs); 6) collect data (CD); 7) send an alert e-mail (SAE); 8) turn on anti-fire, and finally 9) send data to cloud ThingSpeak (CTs) and app ThingView (ATv).

### *The hardware requirements*

Table 2 lists the hardware requirements for implementing the proposed FDAS system, its functionality, and its utilization. The Raspberry Pi 3 *b*+ is a PI series developer board. It is classified as a single-board computer with a LINUX operating system. It has a lot of functions and a fast processing speed, so it is ideal for advanced applications, such as agricultural systems, where it may increase productivity and prevent fires. The choice of the various items of equipment required, as shown in Table 2, is based on a number of technical and functional considerations that are essential to guarantee the effectiveness and reliability of the fire detection and warning system. Firstly, the Raspberry Pi 3 is an optimal choice for this fire detection system due to its balance of processing power, versatility, and strong support for IoT applications, which are critical for ensuring the effective operation of sensor data processing, real-time communication, and cloud integration. Equipped with a quad-core ARM Cortex-A53 processor and 1GB of RAM, the RPi 3 is well-suited for handling simultaneous tasks, such as collecting and processing data from multiple sensors, converting analog inputs via the MCP3008 ADC, and managing MQTT-based communication for seamless data transfer to the cloud. Its multiple GPIO pins provide extensive compatibility with a range of sensors, including flame and smoke detectors, while the MCP3008 ADC ensures accurate analog-to-digital conversion for reliable data acquisition. The Raspberry Pi 3 built-in Wi-Fi and Ethernet connectivity allows for flexible and secure data transmission, supporting protocols such as MQTT that are essential for efficient cloud integration and real-time monitoring on platforms like ThingSpeak. This connectivity further enables the RPi 3 to handle SMTP-based email alerts, which can automatically notify farmers of any detected fires, promoting rapid response. The Raspberry Pi's capability to run Python and access various libraries simplifies the implementation of automated notifications, streamlining both local processing and network communication tasks. Although alternative boards, such as the ESP32, BeagleBone Black, and Arduino with additional shields, could potentially fulfill some system requirements, each presents limitations in multitasking, processing power, or ease of integration compared to the Raspberry Pi 3. The ESP32, while cost-effective and equipped with Wi-Fi and Bluetooth, lacks the Raspberry Pi 3 multitasking capability, which is crucial for managing multiple data streams and cloud communications. Arduino-based alternatives provide basic analog reading and control capabilities but lack the high-level processing required for tasks such as real-time analysis, email alerting, and seamless cloud integration. Consequently, the Raspberry Pi 3′s unique combination of performance, connectivity, and adaptability makes it the most robust and scalable choice for this real-time fire detection application in agriculture, ensuring reliable monitoring, timely alerts, and overall system efficiency.

Secondly, the MCP3208 was selected as the analog-to-digital converter (ADC) because of its ability to process the analog signals emitted by the flame and smoke sensors. This 8-channel, 12-bit resolution device converts analog values into precise digital data, making it easy to interface with the Raspberry Pi. Its use is crucial, as it enables reliable data acquisition and rapid interpretation of detection signals, which is essential for reacting in real-time to emergencies. The flame and smoke sensors were chosen for their sensitivity and specificity in detecting fires. The KY-026 flame sensor effectively detects the infrared radiation emitted by flames, while the MQ-135 smoke sensor is capable of identifying harmful particles and gases associated with combustion. Together, these sensors provide comprehensive coverage for early fire detection, reducing the risk of property damage and injury. In addition, the relay has been integrated into the system to control the activation of extinguishing devices, such as pumps or sprinklers. This component provides electrical isolation between the detection circuit and the extinguishing equipment power supply circuit, guaranteeing safety when fire-fighting systems are activated. On the other hand, the power supply has been carefully selected to provide the necessary energy for the entire system. A stable and adequate power supply is essential to ensure that the sensors, the relay, and the Raspberry Pi work properly. This not only ensures the reliability of the system but also its resilience to voltage fluctuations, which is crucial in emergencies. The hardware components shown in Table 2 were selected on the basis of their performance, compatibility, and ability to work together coherently to provide an effective and responsive fire detection and warning system.

## Results and discussion of the FDAS system

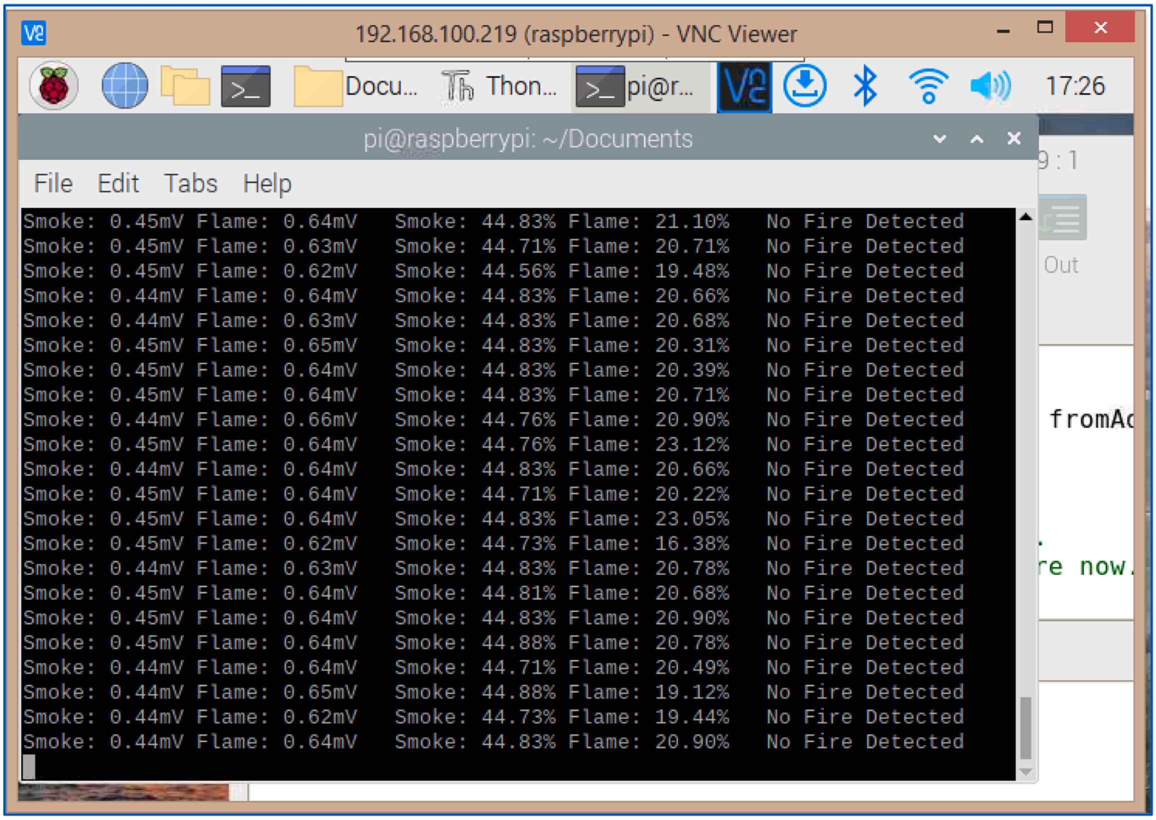
In this section, experiments were carried out to evaluate the efficacy of the proposed fire detection system. To ensure robustness, seven individual alternate situations were considered where each had its own circumstances and difficulties. The goal of this study is to assist farmers in preventing and minimizing the harm that fires can inflict, which can have a considerable effect on agricultural output and food security. Farmers may detect fires in real-time and take prompt action to stop the spread of the fire by having access to a low-cost, simple-to-use, and reliable fire detection system. The innovation of this work is how IoT, embedded systems, and sensors are combined to provide a complete and dependable fire detection system for smart agriculture. The proposed system has four technology levels: 1) the edge network layer, 2) the fog network layer, 3) the cloud computing layer, and 4) the data representation layer. The suggested system makes use of these technologies' strengths to deliver prompt alarms and real-time sensor data monitoring, allowing farmers to respond quickly in the event of a fire. In this section, the implementation of the FDAS system has been made using the ThingSpeak platform and the application ThingView. Moreover, the results of an alert e-mail in the event of a fire by sending an urgent message to the farm owner are processed. The study and visualization of data in real-time by ThingSpeak using online analysis tools are made. The proposed scheme is then compared with the latest scientific paper in the same field.

### *Experimental results of the FDAS system*

The experimental findings of the proposed FDAS system based on the IoT and Raspberry Pi devices are expressed. The values of the sensors on the Raspberry Pi terminal via the VNC viewer are shown in Fig. 6. The graphic shows the smoke sensor value and flame sensor value in mV in the first and second columns, the smoke fire sensor value and flame sensor value in percentage, sensor ID in the last column, and Fire state. Moreover, a background Python program continually enters real fire smoke and flame readings into the database. The analog input code function is the A/D converter theoretical digital output code. The analog input voltage range is determined by the reference input (voltage reference) for each device. The least significant bit (LSB) size decreases as the reference input decreases significantly. The reference input and the input signal [50] demonstrate how the output value is derived from the input and reference voltage using (1). A value of 4095 is utilized while calculating the maximum output value as follows:

$$V_{out} = 4095 \text{ x } V_{in}/V_{ref} \tag{1}$$

where $V_{out}$ is the digital output voltage, $V_{in}$ is the analog input voltage, and $V_{ref}$ is the reference voltage.



```
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.83% Flame: 21.10%   No Fire Detected
Smoke: 0.45mV Flame: 0.63mV   Smoke: 44.71% Flame: 20.71%   No Fire Detected
Smoke: 0.45mV Flame: 0.62mV   Smoke: 44.56% Flame: 19.48%   No Fire Detected
Smoke: 0.44mV Flame: 0.64mV   Smoke: 44.83% Flame: 20.66%   No Fire Detected
Smoke: 0.44mV Flame: 0.63mV   Smoke: 44.83% Flame: 20.68%   No Fire Detected
Smoke: 0.45mV Flame: 0.65mV   Smoke: 44.83% Flame: 20.31%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.83% Flame: 20.39%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.83% Flame: 20.71%   No Fire Detected
Smoke: 0.44mV Flame: 0.66mV   Smoke: 44.76% Flame: 20.90%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.76% Flame: 23.12%   No Fire Detected
Smoke: 0.44mV Flame: 0.64mV   Smoke: 44.83% Flame: 20.66%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.71% Flame: 20.22%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.83% Flame: 23.05%   No Fire Detected
Smoke: 0.45mV Flame: 0.62mV   Smoke: 44.73% Flame: 16.38%   No Fire Detected
Smoke: 0.44mV Flame: 0.63mV   Smoke: 44.83% Flame: 20.78%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.81% Flame: 20.68%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.83% Flame: 20.90%   No Fire Detected
Smoke: 0.45mV Flame: 0.64mV   Smoke: 44.88% Flame: 20.78%   No Fire Detected
Smoke: 0.44mV Flame: 0.64mV   Smoke: 44.71% Flame: 20.49%   No Fire Detected
Smoke: 0.44mV Flame: 0.65mV   Smoke: 44.88% Flame: 19.12%   No Fire Detected
Smoke: 0.44mV Flame: 0.62mV   Smoke: 44.73% Flame: 19.44%   No Fire Detected
Smoke: 0.44mV Flame: 0.64mV   Smoke: 44.83% Flame: 20.90%   No Fire Detected
```

**Fig. 6.** Raspberry Pi terminal.

The MCP3208 converter reads values from 0 as the minimum and 0.99 as the maximum. To make the reading procedure easier for the user or farm owner, a function is utilized to calibrate the sensor value from 0 % to 99 % as a maximum value. For calibration, the numerical value 10 is used so that the sensor values reach from 0 % to 99 %, as shown in (2).

$$Y_{cal} = Y_{ncal} \text{ x } 10 \tag{2}$$

where $Y_{ncal}$ is the non-calibrated output and $Y_{cal}$ is the calibrated output in %.

The test results are shown in Table 3. This table contains the 1) test count, 2) fire smoke sensor value, 3) flame sensor value, 4) fire status, 5) firefighting, 6) sending data to the cloud, 7) sending an e-mail alert, and 8) the time of sending data to the cloud. The accuracy of our proposed system is also shown in the last column which was found to be 100 % accurate for all the tests. The farm had no fire in cases 1, 2, 6, and 7. If there was no fire announcement, the reaction was opposite in cases 3, 4, and 5, where it was a fire, so that the anti-fire system is activated to put out the fire and sent an alert e-mail to the owner of the farm. Whether there is a fire or not on the farm, all data is sent to the cloud ThingSpeak and app ThingView.

The fire smoke sensor and the flame sensor in the suggested system detect the presence or absence of a fire in smart farming. The system detects a fire when the smoke sensor reading is more than 60 %, and the flame sensor reading is more significant than 35 %. This can be seen in Fig. 7 and Table 3 respectively. In the case of a fire and the absence of cayenne in agriculture, Fig. 6 depicts the fire smoke sensor visual and the flame sensor data output.

Utilizing the Putty and VNC viewer interface to simulate the system, together with the cloud ThingSpeak, app ThingView and SMTP browser, to get the results shown in Table 3. The flame sensor and the smoke sensor are two different types of sensors that the fire detection system utilizes to find fires. The readings of the smoke sensor and flame sensor during the testing were noted in Table 3. Analysis of the experimental results revealed in Table 3 provides valuable insights into the effectiveness of the proposed fire detection system. A total of seven tests were carried out at different times, measuring smoke and flame sensor values, fire detection status, activation of the fire detection system, sending data to the cloud via ThingSpeak, sending e-mail alerts via the SMTP protocol, response time, and system accuracy. Tests 1 and 2: The smoke detection values were 44.56 % and 44.58 % respectively, while the flame detection values were 14.25 % and 15.54 %. In these cases, the system correctly identified the absence of fire ('No Fire'), and the anti-fire system remained deactivated (OFF). The data was sent to the cloud, but no e-mail alert was generated. The response time for these tests was 2 s, and accuracy was maintained at 100 %, demonstrating reliable detection. In addition, tests 3, 4, and 5: These tests showed significant detection values, with 64.10 % and 43.66 % for test 3, 73.23 % and 60.14 % for test 4, and 81.23 % and 72.33 % for test 5. In all these cases, the system detected a fire and activated the fire protection system (ON). The data was successfully transmitted to the cloud, and e-mail alerts were also sent to the users concerned. The response time, at a constant 2 s, demonstrates the rapid responsiveness of the system, with 100 % accuracy for these tests. On the other hand, in tests 6 and 7: In these tests, the smoke detection values were 44.92 % and 44.65 %, and the flame detection values were 21.01 % and 17.23 % respectively. As with tests 1 and 2, the system confirmed that there was no fire and kept the fire suppression system in the OFF position. The data was sent to the cloud, but no e-mail alert was generated. Response time remained at 2 s, and accuracy was maintained at 100 %. The experimental results validate the effectiveness and reliability of the proposed fire detection system. The tests demonstrated the system's ability to correctly detect fire situations, activate the appropriate safety measures, and quickly inform users, underlining the potential application of this technology in various environments such as smart homes and agricultural fields.

**Table 2**
The hardware requirements.

| Serial No. | Hardware | Functionality |
|---|---|---|
| | Raspberry Pi 3 b+ | The Raspberry Pi 3 Model B+ is a high-performance development board featuring a 1.4 GHz quad-core 64-bit processor, dual-band connectivity (2.4 GHz/5 GHz Wi-Fi), Bluetooth 4.2/BLE technology and fast Ethernet with Power over Ethernet (PoE) capability. Combining server and microcontroller functions, it is compatible with real-time operating systems, making it ideal for IoT and automation applications requiring high computing power, immediate responsiveness, and real-time data management [49]. |
| | Flame Sensor | The KY-026 flame sensor is a detector sensitive to the infrared radiation emitted by flames, designed to quickly identify the presence of fire in a monitored area. It is equipped with an infrared receiver that detects light variations specific to flames, guaranteeing reliable detection even in partially lit environments. This sensor is particularly well suited to fire safety and real-time warning applications in automated systems. [31]. |
| | Fire Smoke Sensor | The MQ-135 smoke sensor is a highly sensitive gas detector, capable of detecting a variety of noxious gases such as ammonia, carbon dioxide, and hazardous vapors, including combustion smoke. It is widely used for the detection of air pollution and the rapid identification of smoke concentrations, making it an essential component of fire detection systems [39] |
| | MCP3208 | The MCP3208 is a versatile and reliable ADC chip that finds widespread use in the conversion of analog signals to digital data, particularly in applications involving multiple sensors or analog input sources. Its SPI interface, multiple channels, and 12-bit resolution make it a valuable component in a variety of electronic systems. The MCP3208 analog-to-digital converter with Rasberry Pi will be utilized. The MCP3208 converter provides 8 channels of a 12-bit rate [50]. |
| | Relay | A relay was used to control the sprinkler system (pump), which acts as an anti-fire device by activating the extinguishing mechanism as soon as a fire is detected. |
| | Anti-Fire | This including a pump or sprinkler system acts as an extinguisher by spraying water to control the fire and prevent it spreading. |
| | Power Supply | A power supply has been set up to provide the energy required to operate the proposed system, ensuring a stable and continuous supply to all components. |

**Table 3**
Experimental results of the FDAS system.

| SR No. | Fire Smoke Sensor | Flame Sensor | Status | Anti-Fire | Send Data to Cloud | Send Alert E-mail | Response Time | Accuracy |
|---|---|---|---|---|---|---|---|---|
| 1 | 44.56 % | 14.25 % | No Fire | OFF | Yes | No | 2S | 100 % |
| 2 | 44.58 % | 15.54 % | No Fire | OFF | Yes | No | 2S | 100 % |
| 3 | 64.10 % | 43.66 % | Fire | ON | Yes | Yes | 2S | 100 % |
| 4 | 73.23 % | 60.14 % | Fire | ON | Yes | Yes | 2S | 100 % |
| 5 | 81.23 % | 72.33 % | Fire | ON | Yes | Yes | 2S | 100 % |
| 6 | 44.92 % | 21.01 % | No Fire | OFF | Yes | No | 2S | 100 % |
| 7 | 44.65 % | 17.23 % | No Fire | OFF | Yes | No | 2S | 100 % |

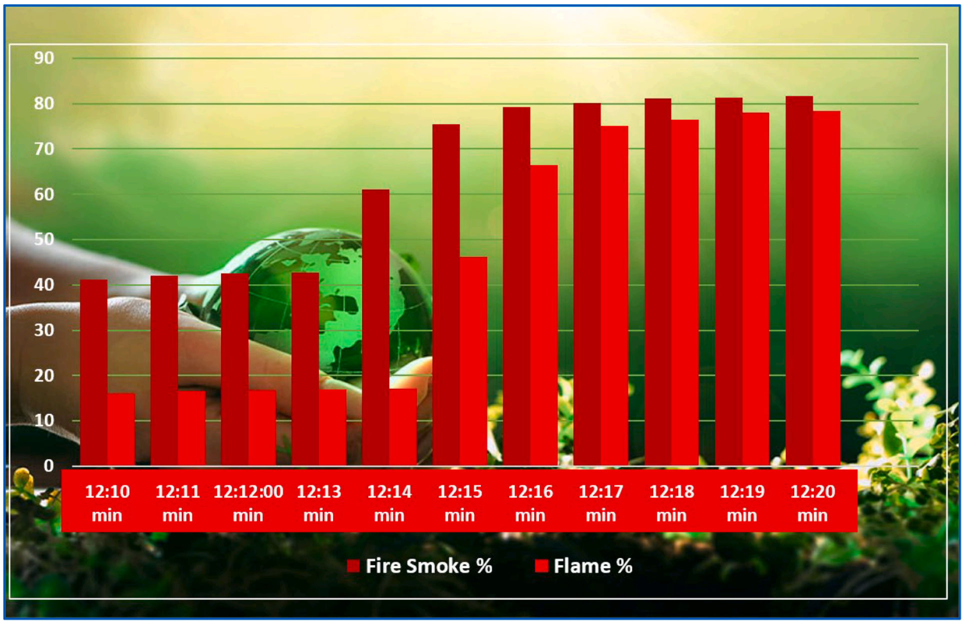


**Fig. 7.** Values of the fire smoke and flame sensor.

**Table 4**
Duration of sending and receiving e-mail notifications.

| Testing | Sending Time (s) | Receiving Time (s) | Round-Trip Time (s) |
|---|---|---|---|
| 1 | 7.10 s | 8.80 s | 1.70 s |
| 2 | 7.30 s | 9.16 s | 1.86 s |
| 3 | 2.04 s | 3.97 s | 1.92 s |
| 4 | 3.18 s | 4.72 s | 1.54 s |

### *Sensors output on the ThingsSpeak cloud and ThingView App*

ThingSpeak is a cloud-based IoT analytics platform that allows to 1) collect, 2) visualize, and 3) analyze the data in real-time. ThingSpeak provides real-time representations of the data obtained from the proposed system. It can perform online analysis and data analysis for the FDAS system as it comes by implementing MATLAB code in ThingSpeak. The data from the fire smoke sensor was shown on a line graph in Fig. 8, along with a numerical value. The x-axis of the line graph represents the date as determined by the data obtained. In contrast, the y-axis represents the range of integers from zero to 100, representing the percentages of smoke levels in smart agriculture. It recorded a percentage in the graph of 49.76 %, and the lowest peak was 21.37 %. The farm owner may now connect to the cloud and use the graphical output to analyze the farm status. Fig. 9 on ThingSpeak shows some flame sensor data in smart agriculture. The flame sensor measurement fluctuates between 25.61 % and 13.86 %. Fig. 10 shows the channel display widgets fire smoke (a) and flame (b) reading.

Fig. 11 demonstrates how to read numerous variables from an existing ThingSpeak channel and create a scatter plot to investigate the correlation between smoke and flame. In this work, the template is changed from one of the MATLAB Analysis to MATLAB

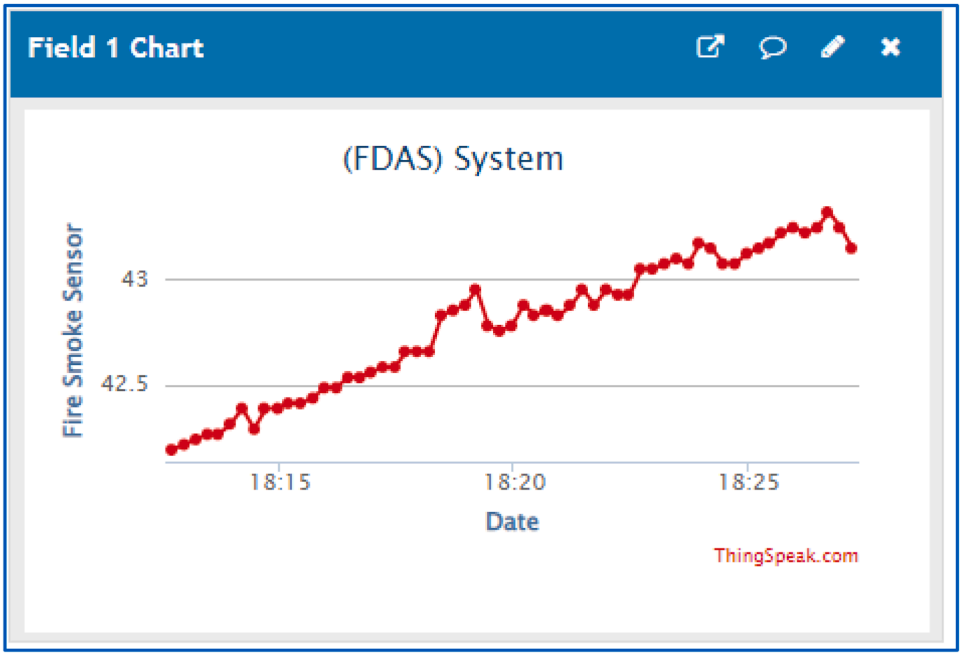


**Fig. 8.** The output of the fire smoke sensor on ThingsSpeak.

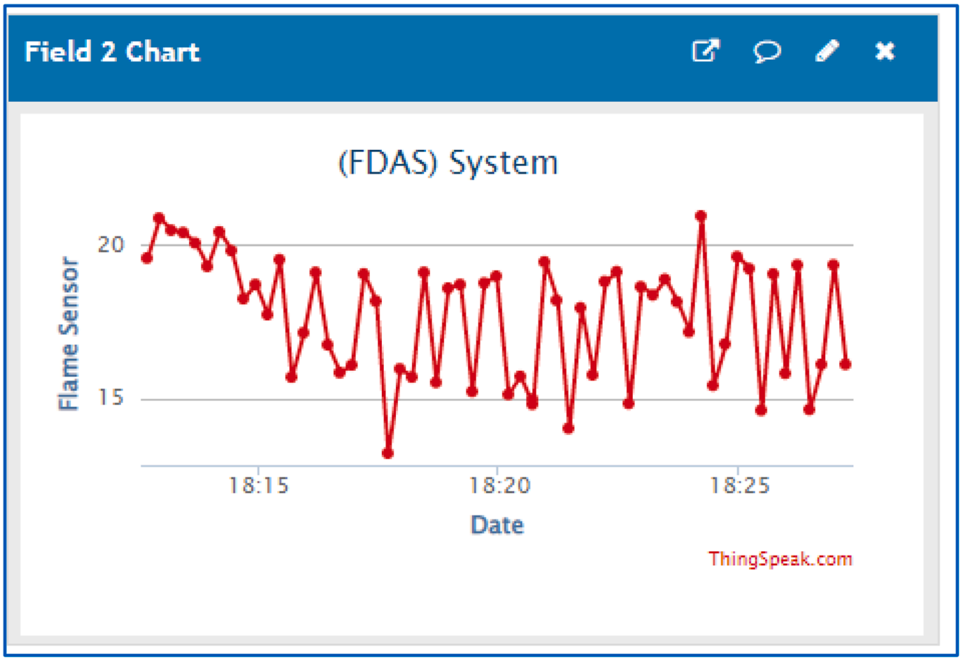
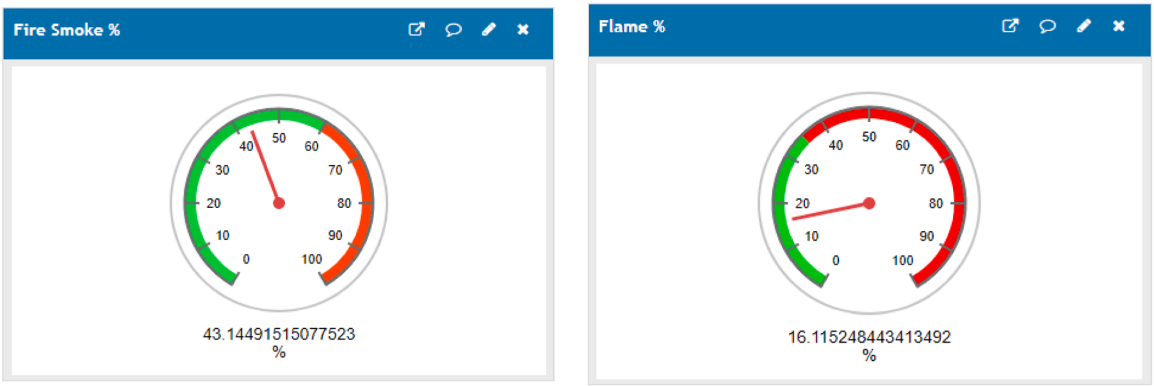


**Fig. 9.** The output of the flame sensor on ThinkSpeak.

Visualizations app code templates. Every 15 s, the Raspberry pi microcontroller uploads the smoke sensor and flame sensor readings to the ThingSpeak IoT analytics platform and plots them. A farmer may use ThingSpeak as an effortless and straightforward way to collect data about the farm in real-time to avoid agricultural losses due to fire. A MATLAB script was developed using the template code provided in MATLAB visualization software to generate a wind speed visualization from the ThingSpeak channel. The wind speed graphic obtained on the same can be seen in Fig. 12.

The location of channels and specific changes to channel content can be tracked. In this location 372, a channel sitemap was established that differed from the feed data location in 373 formations. The "Channel Settings" tab was processed in the channel view.



*(a)* *(b)*

**Fig. 10.** Channel display widgets fire smoke (a) and flame (b) reading.

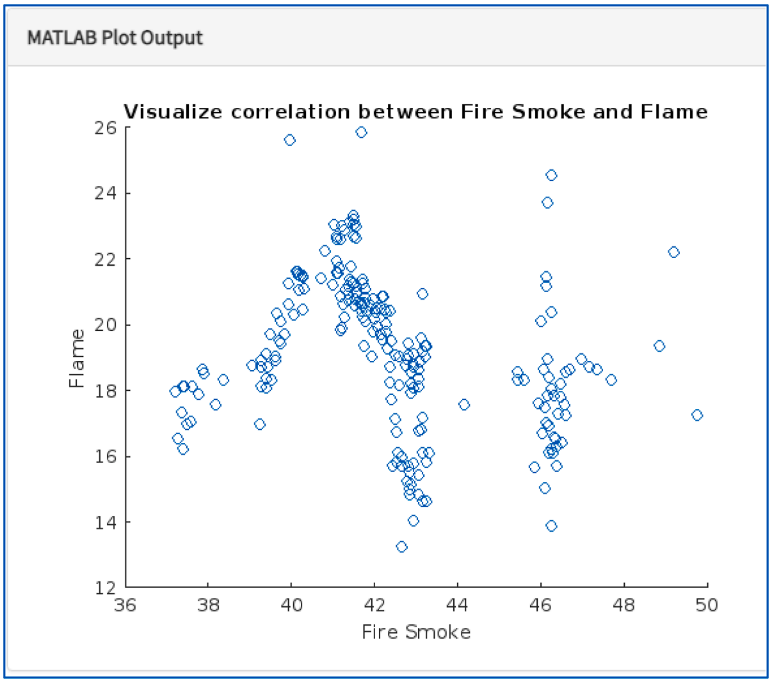


**Fig. 11.** Correlation between fire smoke and flame.

The 374 location was displayed, and the channel latitude and longitude are entered adequately. Fig. 13 shows location 375 and how the channel map is now featured in private and public channel views. Morocco is situated at 376 on the northern frontier of the African continent between latitudes 21°N and 36°N 377 and longitudes 1°W and 17°W. With the help of the Raspberry Pi, the number of sensors is sent directly to the web application over the internet. The values are presented as a graph so that past values can be easily checked, and future values can be expected from the output sensor of a web application (See Fig. 14). Therefore, a farmer uses the ThingView app to get information about the farm directly in the app whether there is a fire on the farm or not.

### *Expert data and send alert E-mail*

ThingSpeak displays measurements in real-time, which may be readily shared and integrated into third-party web apps. The online interface may export data as comma-separated values (CSV) text files. Fig. 15 depicts the sensor results saved in the channel's database on the ThingSpeak platform. There are four columns in Fig. 15. 1) The first column displays the precise time of data collection. 2) The second column was the "entry ID," the data collection sequence. 3) The data collected from the smoke sensor was the third column. 4) The data from the flame sensor was in the final and the fourth column, "field2." All the information was gathered in one package.

Once the system obtained data for an agricultural environment and sent it to the ThingSpeak cloud and the ThingView application,

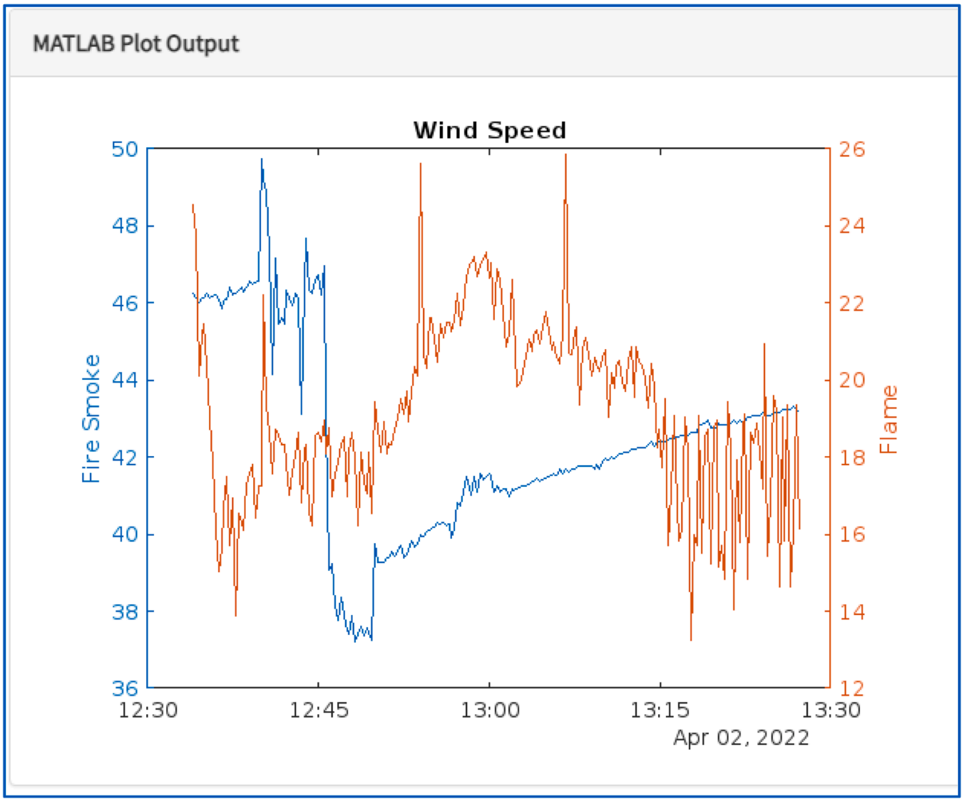


**Fig. 12.** Wind speed.

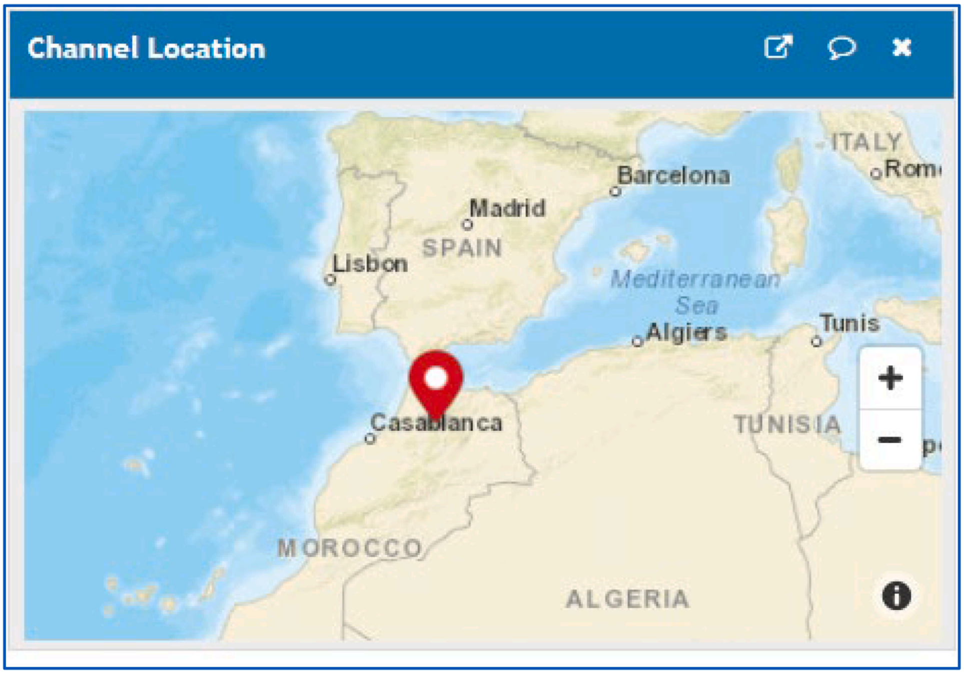
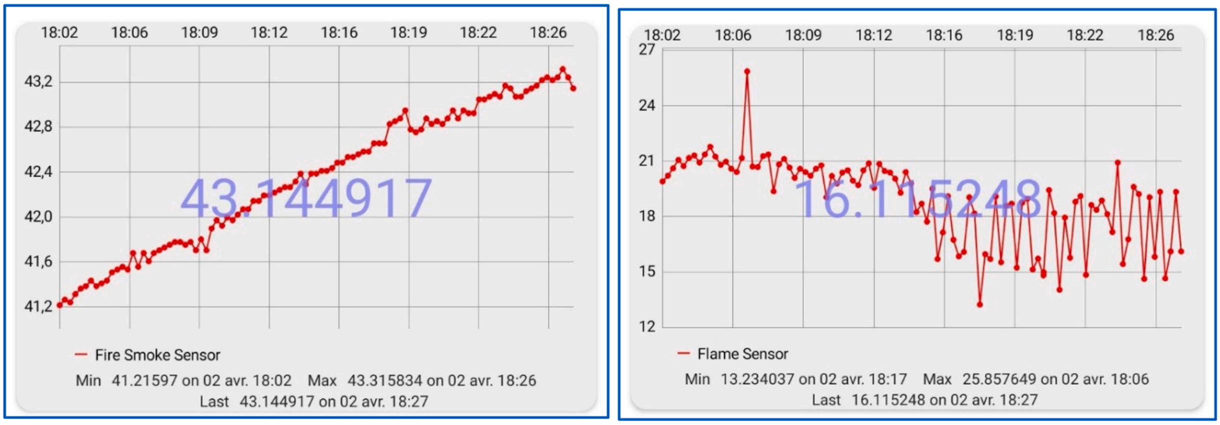
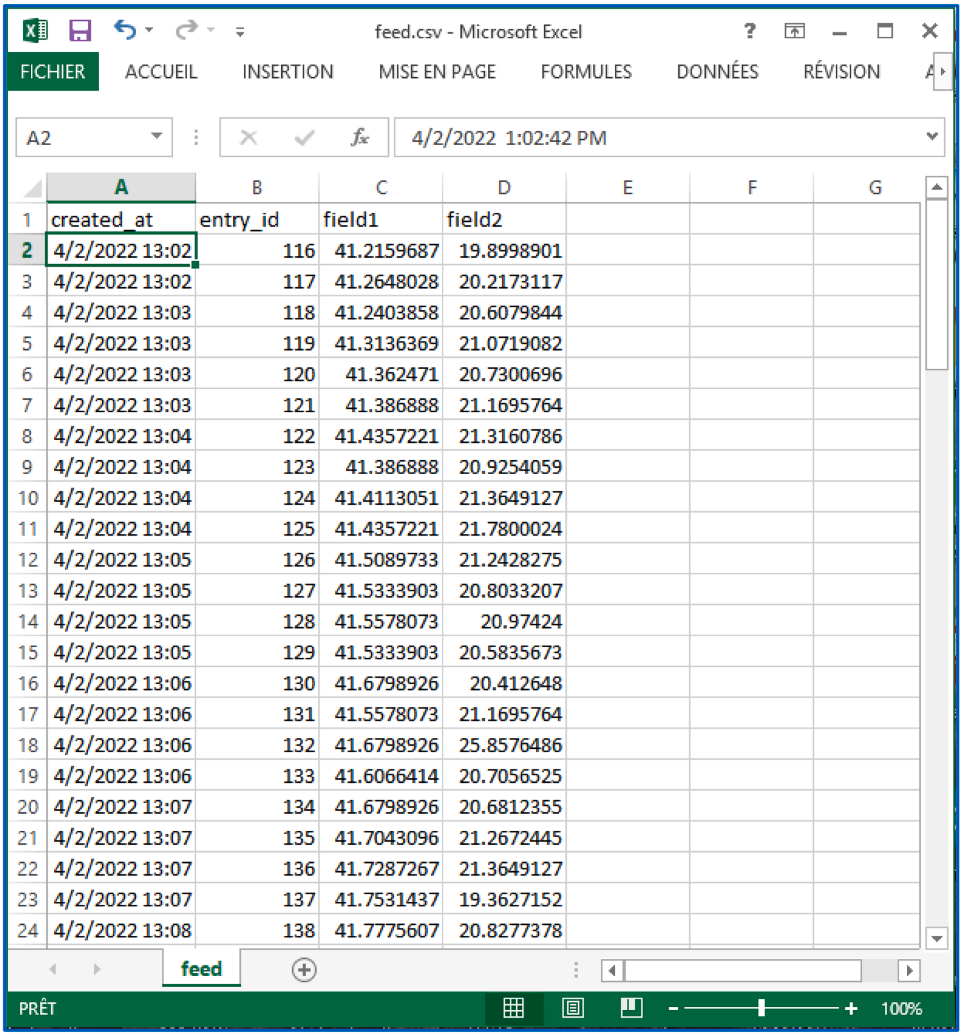


**Fig. 13.** Channel location.



*(a)* *(b)*

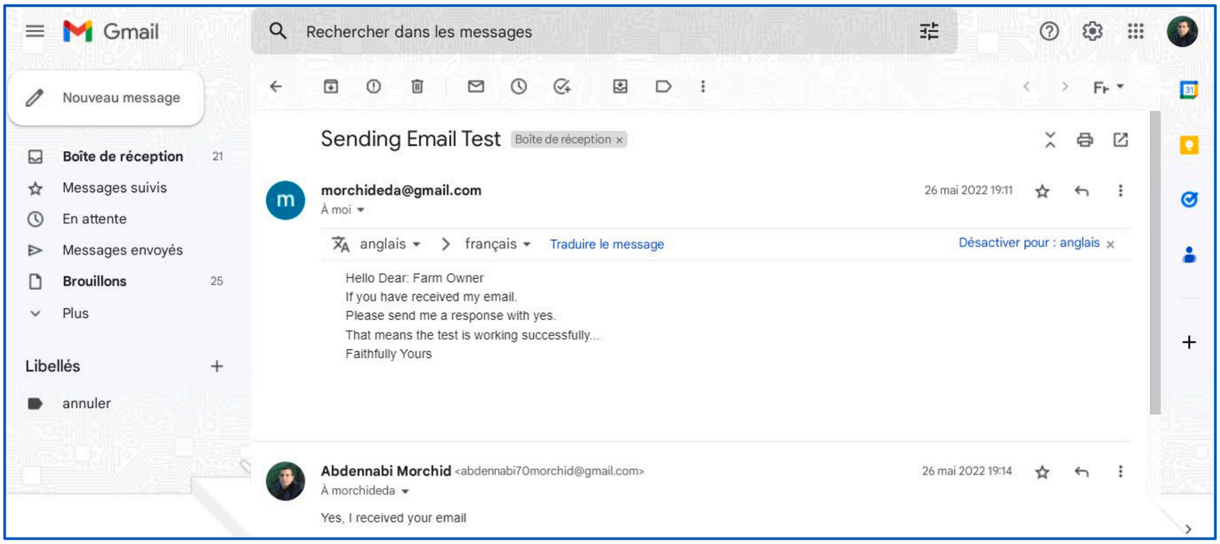
**Fig. 14.** (a) fire smoke sensor, and (b) flame sensor output in ThinkView app in real-time.



A2 | 4/2/2022 1:02:42 PM

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | created_at | entry_id | field1 | field2 | | | |
| 2 | 4/2/2022 13:02 | 116 | 41.2159687 | 19.8998901 | | | |
| 3 | 4/2/2022 13:02 | 117 | 41.2648028 | 20.2173117 | | | |
| 4 | 4/2/2022 13:03 | 118 | 41.2403858 | 20.6079844 | | | |
| 5 | 4/2/2022 13:03 | 119 | 41.3136369 | 21.0719082 | | | |
| 6 | 4/2/2022 13:03 | 120 | 41.362471 | 20.7300696 | | | |
| 7 | 4/2/2022 13:03 | 121 | 41.386888 | 21.1695764 | | | |
| 8 | 4/2/2022 13:04 | 122 | 41.4357221 | 21.3160786 | | | |
| 9 | 4/2/2022 13:04 | 123 | 41.386888 | 20.9254059 | | | |
| 10 | 4/2/2022 13:04 | 124 | 41.4113051 | 21.3649127 | | | |
| 11 | 4/2/2022 13:04 | 125 | 41.4357221 | 21.7800024 | | | |
| 12 | 4/2/2022 13:05 | 126 | 41.5089733 | 21.2428275 | | | |
| 13 | 4/2/2022 13:05 | 127 | 41.5333903 | 20.8033207 | | | |
| 14 | 4/2/2022 13:05 | 128 | 41.5578073 | 20.97424 | | | |
| 15 | 4/2/2022 13:05 | 129 | 41.5333903 | 20.5835673 | | | |
| 16 | 4/2/2022 13:06 | 130 | 41.6798926 | 20.412648 | | | |
| 17 | 4/2/2022 13:06 | 131 | 41.5578073 | 21.1695764 | | | |
| 18 | 4/2/2022 13:06 | 132 | 41.6798926 | 25.8576486 | | | |
| 19 | 4/2/2022 13:06 | 133 | 41.6066414 | 20.7056525 | | | |
| 20 | 4/2/2022 13:07 | 134 | 41.6798926 | 20.6812355 | | | |
| 21 | 4/2/2022 13:07 | 135 | 41.7043096 | 21.2672445 | | | |
| 22 | 4/2/2022 13:07 | 136 | 41.7287267 | 21.3649127 | | | |
| 23 | 4/2/2022 13:07 | 137 | 41.7531437 | 19.3627152 | | | |
| 24 | 4/2/2022 13:08 | 138 | 41.7775607 | 20.8277378 | | | |

**Fig. 15.** Database export.



Sending Email Test

morchideda@gmail.com
À moi
26 mai 2022 19:11

Hello Dear: Farm Owner
If you have received my email.
Please send me a response with yes.
That means the test is working successfully...
Faithfully Yours

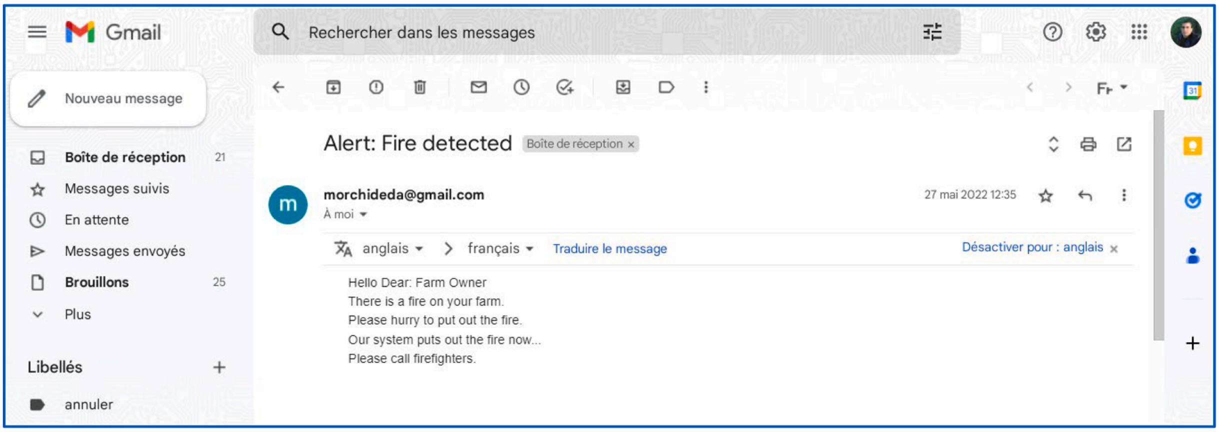
Abdennabi Morchid <abdennabi70morchid@gmail.com>
À morchideda
26 mai 2022 19:14

Yes, I received your email

**Fig. 16.** The alert e-mail test.

an alert e-mail was sent to the farm owner to verify that the message was received, as shown in Fig. 16. To allow access to the SMTP server in Gmail from our app, the Gmail account was logged in with the log-in details. When a fire is detected in smart agriculture, the proposed FDAS system sends an urgent message to the farm owner informing them of a farm fire. This can also be seen in Fig. 17.

The method to send a fire alert email to the farm owner when a fire is discovered is defined in this article to get the time an email takes in milliseconds. The send-time variable is designed to keep track of the email's millisecond-level send time. The receive time variable is designed to hold the time, in milliseconds, at which the email is received after being sent. The email's round-trip time, expressed in milliseconds, is then used to determine the round-trip time variable, as illustrated in Fig. 18. The round-trip time is equal to the difference between the time it is received and the time it is sent. The internet throughput, including the Wi-Fi connection's speed, affects the round-trip time for sending an alert email. The email will get to the farm owner faster since a better internet connection will often result in a shorter round-trip time. It is important to keep in mind that the round-trip time can also be impacted by variables like network congestion, server load, and email message size. Yet, as SMTP protocol is a highly quick technology, a faster internet connection will typically lead to quicker email delivery times. In order to guarantee that alarm emails are issued and received as promptly as possible in the event of fire detection in smart agriculture, it is crucial to ensure that the system is using a dependable and speedy internet connection. Testing the time between sending and receiving e-mails when a fire is detected in smart farming. The SMTP protocol sends and receives e-mails, as shown in Table 4. The following results can be inferred from the test results given in Table 4. When the fire is on the farm, the proposed system takes about 1.75 s to send and receive e-mail alerts as shown in the four tests. Since the Raspberry Pi must always be connected to the internet, these findings are significant in the time because the system uses an anti-fire system when it detects a fire. It also sends an e-mail alert warning of the presence of a fire in a short period allowing firefighters to respond quickly and extinguish the fire and avoid agricultural and human losses. One advantage of the e-mail alert is that this information may be sent to 1) farm owners, 2) workers, and 3) firefighters, which is a massive benefit in smart farming.

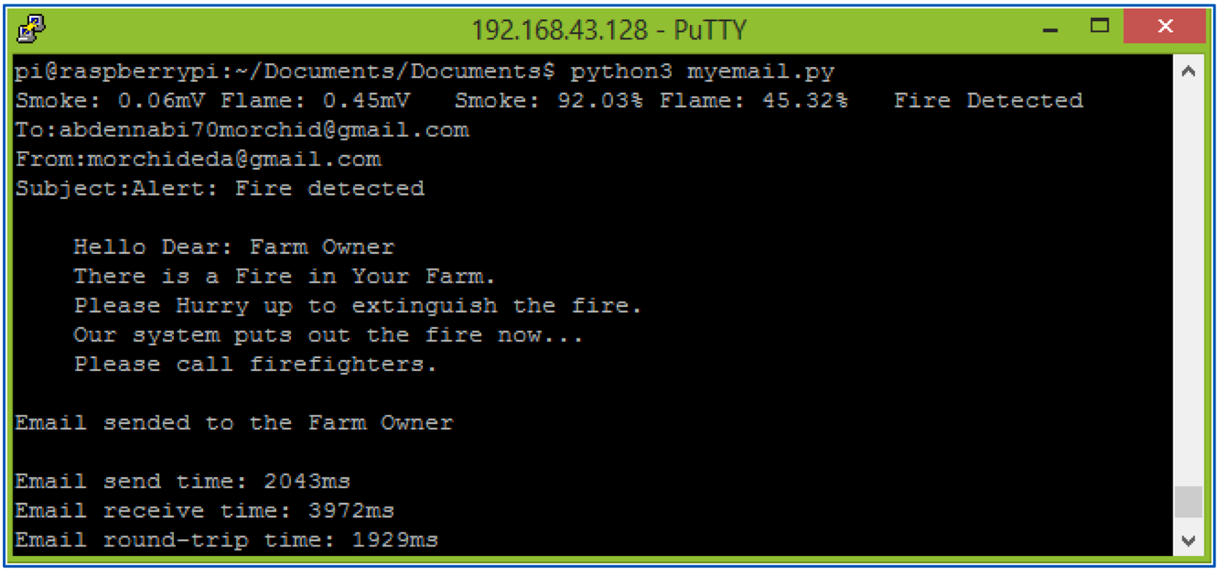


Alert: Fire detected

morchideda@gmail.com
À moi
27 mai 2022 12:35

Hello Dear: Farm Owner
There is a fire on your farm.
Please hurry to put out the fire.
Our system puts out the fire now...
Please call firefighters.

**Fig. 17.** Alert e-mail.



```
pi@raspberrypi:~/Documents/Documents$ python3 myemail.py
Smoke: 0.06mV Flame: 0.45mV   Smoke: 92.03% Flame: 45.32%   Fire Detected
To:abdennabi70morchid@gmail.com
From:morchideda@gmail.com
Subject:Alert: Fire detected

    Hello Dear: Farm Owner
    There is a Fire in Your Farm.
    Please Hurry up to extinguish the fire.
    Our system puts out the fire now...
    Please call firefighters.

Email sended to the Farm Owner

Email send time: 2043ms
Email receive time: 3972ms
Email round-trip time: 1929ms
```

**Fig. 18.** Test alert e-mail round trip time on PuTTy terminal.

**Table 5**
Comparison of related research[5].

| **Paper ID** | **Year** | **Platform Cloud** | Microcontroller with Wi-fi module | **Data Analytics** | **Communication Protocol** | **Send Alert E-mail** | **Sensor Calibration (Threshold Selection)** | **Programming Language** |
|---|---|---|---|---|---|---|---|---|
| [28] | 2019 | No | Arduino Uno | No | No | No | No | Arduino IDE |
| [30] | 2021 | Ubidots platform | ESP32 | No | No | No | No | C++ |
| [32] | 2021 | Web page | Arduino Mega 2560 and ESP8266 | No | Yes (HTTP) | Yes (SMS) | No | Arduino IDE |
| [34] | 2021 | No | Arduino Uno | No | No | No | No | Arduino IDE |
| [39] | 2020 | Web page | Arduino Mega, ESP8266 and nRF24L01 | No | Yes (HTTP) | Yes (SMS) | No | Arduino IDE |
| PS | 2024 | ThingSpeak Cloud And Android Application (ThingView) | Raspberry Pi 3 *b*+ | Yes | Yes (MQTT and SMTP) | Yes (Alert email) | Yes | Python |

[5] PS, SMTP, MQTT, and IDE are the acronyms of the proposed scheme, simple mail transfer protocol and message queue telemetry transport, and integrated development environment, respectively.

### *Discussion and comparison*

The authors of the research [12] emphasize that fundamental information on the causes and consequences of forest loss might aid managers in understanding current trends brought on by anthropogenic climate change and human activity. By comparing forest loss and fires across several years, it is possible to determine that intense wildfires are the direct cause of roughly half of the forest loss. The authors of the research [14] provided a literature evaluation that was done to examine how digital agricultural technologies may prevent/reduce food loss and waste from a global viewpoint, or how little they can do so. The authors of the work [15] provide a thorough analysis of contemporary methods for detecting forest fires that are based on cutting-edge routing protocols such as proactive optimal link state routing, reactive location-aided routing, and dependable routing protocol. The authors of the article [16] suggest an edge computing-focused approach that makes use of the idea of mobile hubs. The created IoT fire detection prototype application employs event processing agents that operate on the smartphones that forest guards carry and is based on the ContextNet middleware. In this study [17], the design and implementation of an IoT-based system that can forecast and detect forest fires and notify the precise position of concerned authorities to aid firefighting personnel in putting the fire out in the area are presented. To offer a better fire inspection solution, the authors of the study [25] concentrate on real-time detection and abnormal condition prediction using a collection of wireless equipment of fire water systems. This study's major objective is to increase fire extinguishing effectiveness. Rescue and firefighting efforts are essential, and hidden or dry fire hydrants may result in firefighting failures. This study [26] implements a smart fire detection system using sensors like smoke sensors and flame sensors that simultaneously protects people and valuable goods by alerting property owners, emergency services, and neighborhood police stations to the fire. In this work [27], a dataset of actual forest fires was utilized to identify the real-time behavior of IoT sensors. This will facilitate higher operational efficiency, throughput, and secure field network management decisions for the precision agricultural sector. According to this study [31], a fire safety system is developed and put into use using a new embedded board that is based on the Atmega328p microcontroller to handle fire control and extinguishing at the level of smart farms. The recommended board runs and monitors itself (automatic maintenance of the siren and the sprinklers). The suggested fire prevention system of the new integrated board consists of a voltage sensor, an H-bridge, an ATMega328p microcontroller, a flame sensor, an MQ-135 gas sensor, a water sprinkler, and an electronic relay. The system is simulated via the EasyEDA (electronic design automation) online software tool.

Although earlier research on fire detection produced useful findings, it seldom used modern technology like the IoT, embedded systems, and sensors to gather data and transfer it to the cloud to remotely monitor the farm via a platform or application. The proposed system has four technology levels: 1) the edge network layer, 2) the fog network layer, 3) the cloud computing layer, and 4) the data representation layer. These technologies have the potential to significantly improve the precision, responsiveness, and usability of fire detection systems. By using the architecture of IoT and embedded systems, this paper's fire detection system can collect data from sensors and send it to the cloud for analysis and display on a platform or application in real-time. The system also makes use of alert email to quickly notify the farm's owner so they have plenty of time to take appropriate action. The study also presents sensor calibration, which is important for guaranteeing that the data gathered is precise and simple for users to comprehend. Overall, the application of modern technology in this system has the potential to significantly improve agricultural fire detection and prevention, which is what distinguishes this study as more unique and innovative.

Table 5 compares the results of the current papers with our proposed FDAS system. The different vital features of systems are compared, such as 1) platform cloud, 2) hardware of the Wi-Fi controller, 3) data analytics, 4) protocol communication, 5) ADC converter, 6) e-mail alert function, 7) sensor calibration (threshold selection), and the 8) programming language used to program systems. The scientific papers presented in the table relate to firefighting systems. These scientific papers differ in terms of the conclusions reached by the researchers. The authors of the research [28] and [34] did not use the IoT to send data to the cloud. Thus, the data was not observed or processed using the e-mail alert. The same is true for the authors of the paper [30]. It was expected to utilize the Ubidots platform. This platform has minimal advantages so that the farm owner can only observe and not analyze or deal with the data. The authors of the papers [32] and [39] used a cloud (web page). However, this cloud has minimal advantages, so the data can only be observed and cannot be analyzed or dealt with by the farm owner. Note that not all searches provide sensor calibration. The work has provided sensor calibration to make it easier to read values for users. On the other hand, the ThingSpeak Cloud and Android Application (ThingView) were utilized in the proposed system. This platform enables to monitor and analyze the data using the features of the MATLAB (MathWorks) lab. The data can also be downloaded. In addition to that, an e-mail alert was sent using the SMTP protocol. Most of the other researchers used the Arduino IDE programming language which is a limited language in terms of features. The Python language, Python language was utilized for programming the research. Python is an excellent language, especially for IoT issues. The Raspberry Pi 3 version B units were utilized because of the provision of extra features. The processor of these units is very efficient with 4 cores and high speed. There is no need to add Wi-Fi or Bluetooth modules besides the low cost which is critical to the cost of the proposed system, unlike most of the Arduino limited performance.

The findings demonstrate that a cloud-based IoT system efficiently: 1) collects, 2) stores, 3) analyzes, and 4) presents environmental data for the proposed system. When a farm fire is detected, the fire detection and anti-fire security FDAS system automatically sends e-mail notifications. An e-mail message takes an average of seconds to send. This is considered a viable solution. To eliminate the fire problem diagnosed by the study [5] done by the UNEP and GRID-Arendal, where it was found that climate change and the use of climate change are projected to contribute to increased fire risk. Wildfires are expected to grow in frequency and severity with a global increase of up to 14 % by 2030, 30 % by the end of 2050, and 50 % by the end of the century. Thus, the proposed system is characterized by effective results at a lower cost.

## Conclusion and future work

This paper proposed an FDAS system using the IoTs in smart agriculture. The proposed system uses an embedded system like a Raspberry Pi. Moreover, sensors were also utilized, such as a fire smoke sensor to measure the amounts of fire smoke in the air and a flame sensor to measure the percentage of fire in an area. The proposed system is more efficient, robust, and reliable. This system can be implemented in all areas, the most important of which are areas that include oases, agriculture, and the forest. The proposed system has valuable and easy-to-use features using a computer capable of real-time processing and sending data via the IoTs. In addition to that, the sensors are more 1) sensitive, 2) cost-effective, and 3) reliable. In general, these sensors are more effective than conventional fire detection systems. The online sensor data was sent to the ThingSpeak cloud via MQTT for further visualization and analysis. The available data is stored, processed, and visualized by the ThingSpeak platform in real-time. The data was also sent to an application, ThingView. An e-mail alert is sent to the farm owner if a fire is detected using the SMTP protocol. When a fire is detected, the proposed system operates at record speed to put out fires at record speed with an average e-mail sending time of 1.75 s. The data was prepared, filtered, and analysed using the MATLAB analysis application and the Python programming language to develop the source code. The performance results of the proposed scheme show an accurate fire detection and anti-fire system performance for improving food security and sustainability in agriculture. An essential set of results was obtained and compared to other scientific papers, especially since studies are confirming that the rate of fire outbreaks will rise in the coming years. The study utilized new technologies such as the IoTs, embedded systems, sensors, and communication protocols. The study will reduce fire hazards in smart agriculture and have many benefits, including 1) preserving green fields, 2) crops, 3) lives, 4) improved agricultural sustainability, 5) ensuring food security for the global population, and hence 6) an improved operational efficiency and productivity. In future work, it is aimed to focus on temperature monitoring and analyze its variants in different regions globally. This could be a major contributing parameter to the efficiency optimization of the firefighting system. This will be further canvassed by utilizing embedded systems-based IoT and image processing for fire detection in the green world and smart farming.

## Author contribution statement

**A. Morchid:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – original draft. **I. G. M. Alblushi:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **H. M Khalid:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **R. El Alami:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **Z. Said:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **H. Qjidaa:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **E. Cuce:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **S. M. Muyeen:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing. **M. O. Jamil:** Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents; Materials; Analysis tools or Data; Writing – review & editing.

## Funding statement

There is no funding for this publication.

### *Data availability statement*

Data will be made available on request.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] Vereinte Nationen, Global Population Growth and Sustainable Development, ed., United Nations, New York, 2021.

[2] Food and Agriculture Organization of the United Nations, The Future of Food and agriculture: Trends and Challenges, ed., Food and Agriculture Organization of the United Nations, Rome, 2017.

[3] E. Fukase, W. Martin, Economic growth, convergence, and world food demand and supply, World Dev. 132 (2020) 104954, https://doi.org/10.1016/j.worlddev.2020.104954.

[4] C. By-Nc-Sa, The Future of Food and Agriculture – Alternative pathways to 2050, Food and Agriculture Organization of the United Nations, Rome, 2018 (n.d.).

[5] Spreading Like wildfire: The rising Threat of Extraordinary Landscape fires, United Nations Environment Programme, A UNEP Rapid Response, Assessment, Nairobi, 2022.

[6] D. Phillips, A. Wasley, A. Heal, D. Camargos, Revealed: Rampant Deforestation of Amazon driven By Global Greed For Meat, The Guardian, 2019. https://www.theguardian.com/environment/2019/jul/02/revealed-amazon-deforestation-driven-global-greed-meat-brazil. accessed October 24, 2024.

[7] P. Vale, H. Gibbs, R. Vale, M. Christie, E. Florence, J. Munger, D. Sabaini, The Expansion of Intensive Beef Farming to the Brazilian Amazon, Global Environmental Change 57 (2019) 101922, https://doi.org/10.1016/j.gloenvcha.2019.05.006.

[8] Q. Sun, C. Miao, M. Hanel, A.G.L. Borthwick, Q. Duan, D. Ji, H. Li, Global heat stress on health, wildfires, and agricultural crops under different levels of climate warming, Environ. Int. 128 (2019) 125–136, https://doi.org/10.1016/j.envint.2019.04.025.

[9] V.R. Pivello, I. Vieira, A.V. Christianini, D.B. Ribeiro, L. da Silva Menezes, C.N. Berlinck, F.P.L. Melo, J.A. Marengo, C.G. Tornquist, W.M. Tomas, G.E. Overbeck, Understanding Brazil's catastrophic fires: causes, consequences and policy needed to prevent future tragedies, Perspect. Ecol. Conserv. 19 (2021) 233–255, https://doi.org/10.1016/j.pecon.2021.06.005.

[10] S.K.-M.W. News, Fires burn over 200 hectares of forest in Morocco's Chefchaouen, (n.d.). https://www.moroccoworldnews.com/2021/08/343922/fires-burn-over-200-hectares-of-forest-in-moroccos-chefchaouen. (accessed October 24, 2024).

[11] O. Jellouli, A.S. Bernoussi, The impact of dynamic wind flow behavior on forest fire spread using cellular automata: application to the watershed BOUKHALEF (Morocco), Ecol. Modell. 468 (2022) 109938, https://doi.org/10.1016/j.ecolmodel.2022.109938.

[12] I. Castro, A.B. Stan, L. Taiqui, E. Schiefer, A. Ghallab, M. Derak, P.Z. Fulé, Detecting fire-caused forest loss in a Moroccan protected area, Fire 5 (2022) 51, https://doi.org/10.3390/fire5020051.

[13] P. Pingali, A. Aiyar, M. Abraham, A. Rahman, Agricultural technology for increasing competitiveness of small holders, in: 2019. https://doi.org/10.1007/978-3-030-14409-8_9.

[14] A. (Addis) Benyam, T. Soma, E. Fraser, Digital agricultural technologies for food loss and waste prevention and reduction: global trends, adoption opportunities and barriers, J. Clean. Prod. 323 (2021) 129099, https://doi.org/10.1016/j.jclepro.2021.129099.

[15] F.T. AL-Dhief, N. Sabri, S. Fouad, N.M.A. Latiff, M.A.A. Albader, A review of forest fire surveillance technologies: mobile ad-hoc network routing protocols perspective, J. King Saud University - Comput. Inf. Sci. 31 (2019) 135–146, https://doi.org/10.1016/j.jksuci.2017.12.005.

[16] G.B. Neumann, V.P. de Almeida, M. Endler, Smart Forests: fire detection service, in: 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, pp. 01276–01279, https://doi.org/10.1109/ISCC.2018.8538719.

[17] K. Jayaram, K. Janani, R. Jeyaguru, R. Kumaresh, N. Muralidharan, Forest Fire Alerting System With GPS Co-ordinates Using IoT, in: 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 488–491, https://doi.org/10.1109/ICACCS.2019.8728383.

[18] J. Xu, B. Gu, G. Tian, Review of agricultural IoT technology, Artificial Intell. Agricult. 6 (2022) 10–22, https://doi.org/10.1016/j.aiia.2022.01.001.

[19] R. Sharma, B. Villányi, A sustainable Ethereum merge-based Big-Data gathering and dissemination in IIoT System, Alexandria Eng. J. 69 (2023) 109–119, https://doi.org/10.1016/j.aej.2023.01.055.

[20] M. Gupta, M. Abdelsalam, S. Khorsandroo, S. Mittal, Security and privacy in smart farming: challenges and opportunities, IEEE Access. 8 (2020) 34564–34584, https://doi.org/10.1109/ACCESS.2020.2975142.

[21] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, E.-H.M. Aggoune, Internet-of-Things (IoT)-Based Smart Agriculture: toward Making the Fields Talk, IEEE Access. 7 (2019) 129551–129583, https://doi.org/10.1109/ACCESS.2019.2932609.

[22] J. Chen, A. Yang, Intelligent Agriculture and Its Key Technologies Based on Internet of Things Architecture, IEEe Access. 7 (2019) 77134–77141, https://doi.org/10.1109/ACCESS.2019.2921391.

[23] M.Esmail Karar, A.-H. Abdel-Aty, F. Algarni, M. Fadzil Hassan, M.A. Abdou, O. Reyad, Smart IoT-based system for detecting RPW larvae in date palms using mixed depthwise convolutional networks, Alexandria Eng. J. 61 (2022) 5309–5319, https://doi.org/10.1016/j.aej.2021.10.050.

[24] The Impact of Disasters and Crises On Agriculture and Food security: 2021, FAO, 2021, https://doi.org/10.4060/cb3673en.

[25] J. Wu, J. Mao, J. Hu, X. Shu, Monitoring and prediction of fire water system based on wireless detection method, Fire Technol. 58 (2022) 2213–2228, https://doi.org/10.1007/s10694-022-01235-0.

[26] H. Alqourabah, A. Muneer, S. Fati, A smart fire detection system using IoT technology with automatic water sprinkler, Int. J. Electr. Comput. Eng. (2020), https://doi.org/10.11591/ijece.v11i4.pp2994-3002.

[27] K. Sood, M.R. Nosouhi, N. Kumar, A. Gaddam, B. Feng, S. Yu, Accurate detection of IoT sensor behaviors in legitimate, faulty and compromised scenarios, IEEE Trans. Dependable Secure Comput. 20 (2023) 288–300, https://doi.org/10.1109/TDSC.2021.3131991.

[28] B. Sarwar, I.S. Bajwa, N. Jamil, S. Ramzan, N. Sarwar, An intelligent fire warning application using iot and an adaptive neuro-fuzzy inference system, Sensors 19 (2019) 3150, https://doi.org/10.3390/s19143150.

[29] N.A. Zakaria, Z.Z. Abidin, N. Harum, L.C. Hau, N.S. Ali, F.A. Jafar, Wireless internet of things-based air quality device for smart pollution monitoring, Int. J. Adv. Comput. Sci. Appl. 9 (n.d.). https://doi.org/10.14569/IJACSA.2018.091110.

[30] P. Kanakaraja, P. Syam Sundar, N. Vaishnavi, S. Gopal Krishna Reddy, G. Sai Manikanta, IoT enabled advanced forest fire detecting and monitoring on Ubidots platform, Mater. Today 46 (2021) 3907–3914, https://doi.org/10.1016/j.matpr.2021.02.343.

[31] A. Morchid, R. El Alami, Q. Hassan, M. Alaoui, H. Touati, Mohammed Ouazzani Jamil, Fire safety system implementation for controlling and monitoring a siren in smart farm using gas sensor and flame sensor, 2022. https://doi.org/10.1007/978-3-031-01942-5_73.

[32] M. Mahbub, M.M. Hossain, Md.S.A. Gazi, Cloud-Enabled IoT-based embedded system and software for intelligent indoor lighting, ventilation, early stage fire detection and prevention, Comput. Netw. 184 (2021) 107673, https://doi.org/10.1016/j.comnet.2020.107673.

[33] C. Xiong, Z. Wang, Y. Huang, F. Shi, X. Huang, Smart evaluation of building fire scenario and hazard by attenuation of alarm sound field, J. Build. Eng. 51 (2022) 104264, https://doi.org/10.1016/j.jobe.2022.104264.

[34] A. Rehman, M.A. Qureshi, T. Ali, M. Irfan, S. Abdullah, S. Yasin, U. Draz, A. Glowacz, G. Nowakowski, A. Alghamdi, A.A. Alsulami, M. Węgrzyn, Smart Fire detection and deterrent system for human savior by using Internet of Things (IoT), Energies. (Basel) 14 (2021) 5500, https://doi.org/10.3390/en14175500.

[35] A. Sharma, P.K. Singh, Y. Kumar, An integrated fire detection system using IoT and image processing technique for smart cities, Sustain. Cities. Soc. 61 (2020) 102332, https://doi.org/10.1016/j.scs.2020.102332.

[36] C. Maraveas, D. Loukatos, T. Bartzanas, K.G. Arvanitis, Applications of artificial intelligence in fire safety of agricultural structures, Appl. Sci. 11 (2021) 7716, https://doi.org/10.3390/app11167716.

[37] T. Seesaard, N. Goel, M. Kumar, C. Wongchoosuk, Advances in gas sensors and electronic nose technologies for agricultural cycle applications, Comput. Electron. Agric. 193 (2022) 106673, https://doi.org/10.1016/j.compag.2021.106673.

[38] R.A. Sowah, A.R. Ofoli, S.N. Krakani, S.Y. Fiawoo, Hardware design and web-based communication modules of a real-time multisensor fire detection and notification system using fuzzy logic, IEEE Trans. Ind. Appl. 53 (2017) 559–566, https://doi.org/10.1109/TIA.2016.2613075.

[39] M. Mahbub, A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network, Internet Things 9 (2020) 100161, https://doi.org/10.1016/j.iot.2020.100161.

[40] A. Morchid, M. El Alaoui, R. El Alami, H. Qjidaa, K. El Khadiri, Y. Mehdaoui, Design and realization of fire safety system for controlling and monitoring a siren using Arduino Uno, in: 2021. https://doi.org/10.1007/978-3-030-73882-2_40.

[41] A. Dudnik, P. Daria, M. Kobylchuk, T. Domkiv, N. Dahno, O. Leshchenko, Intrusion and fire detection method by wireless sensor network, in: 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT), 2020, pp. 211–215, https://doi.org/10.1109/ATIT50783.2020.9349351.

[42] B. Mishra, A. Kertesz, The Use of MQTT in M2M and IoT Systems: A Survey, 8, IEEE Access, 2020, pp. 201071–201086, https://doi.org/10.1109/ACCESS.2020.3035849.

[43] A. Morchid, I.G. Muhammad Alblushi, H.M. Khalid, R. El Alami, S.R. Sitaramanan, S.M. Muyeen, High-technology agriculture system to enhance food security: a concept of smart irrigation system using Internet of Things and cloud computing, J. Saudi Soc. Agricult. Sci. (2024), https://doi.org/10.1016/j.jssas.2024.02.001.

[44] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, M. Alazab, A comprehensive survey for intelligent spam email detection, IEEE Access. 7 (2019) 168261–168295, https://doi.org/10.1109/ACCESS.2019.2954791.

[45] R. Doraiswami, L. Cheded, Echo cancellation: a novel adaptive Kalman filter-based scheme, 2016. https://doi.org/10.11159/cdsr16.112.

[46] A. Morchid, M. Marhoun, R. El Alami, B. Boukili, Intelligent detection for sustainable agriculture: a review of IoT-based embedded systems, cloud platforms, DL, and ML for plant disease detection, Multimed. Tools. Appl. 83 (2024) 70961–71000, https://doi.org/10.1007/s11042-024-18392-9.

[47] D.A. Vinnik, A.Yu. Starikov, V.E. Zhivulin, K.A. Astapovich, V.A. Turchenko, T.I. Zubar, S.V. Trukhanov, J. Kohout, T. Kmječ, O. Yakovenko, L. Matzui, A.S. B. Sombra, D. Zhou, R.B. Jotania, C. Singh, Y. Yang, A.V. Trukhanov, Changes in the Structure, Magnetization, and Resistivity of BaFe12–xTixO19, ACS Appl. Electron. Mater 3 (2021) 1583–1593, https://doi.org/10.1021/acsaelm.0c01081.

[48] Y. Yao, V.E. Zhivulin, A.R. Zykova, N.A. Cherkasova, D.A. Vinnik, E.A. Trofimov, S.A. Gudkova, O.V. Zaitseva, S.V. Taskaev, L.N. Alyabyeva, B.P. Gorshunov, A. A. Gurchenko, S. Lu, S.V. Trukhanov, A.V. Trukhanov, High entropy BaFe12-x(Ti/Mn/Ga/In)xO19 (x = 1–7) oxides: correlation of the composition, entropy state, magnetic characteristics, and terahertz properties, Ceram. Int. 49 (2023) 31549–31558, https://doi.org/10.1016/j.ceramint.2023.07.106.

[49] A. Morchid, R. Jebabra, A. Ismail, H.M. Khalid, R. El Alami, H. Qjidaa, M.Ouazzani Jamil, IoT-enabled fire detection for sustainable agriculture: a real-time system using flask and embedded technologies, Results. Eng. 23 (2024) 102705, https://doi.org/10.1016/j.rineng.2024.102705.

[50] 2.7V 4-Channel/8-Channel 12-Bit A/D Converters with SPI Serial Interface, (2008).