Received: 23 January 2023 | Revised: 12 March 2023 | Accepted: 12 April 2023

IET Cyber-Physical Systems: Theory & Applications

DOI: 10.1049/cps2.12054

The Institution of Engineering and Technology WILEY

CASE STUDY

Digital farming based on a smart and user-friendly IoT irrigation system: A conifer nursery case study

Adrian Florea[1] | Daniel-Ioan Popa[1] | Daniel Morariu[1] | Ionela Maniu[2] | Lasse Berntzen[3] | Ugo Fiore[4]

[1]Department of Computer Science and Electrical Engineering, Lucian Blaga University of Sibiu, Sibiu, Romania

[2]Department of Mathematics and Informatics, Research Center in Informatics and Information Technology, Lucian Blaga University of Sibiu, Sibiu, Romania

[3]School of Business, University of South-Eastern Norway, Borre, Norway

[4]Department of Computer Science, University of Salerno, Fisciano, Italy

Correspondence

Adrian Florea.
Email: adrian.florea@ulbsibiu.ro

Funding information

MAPIoT, Grant/Award Number: 20-COP-0019; EEA Financial Mechanism 2014–2021

Abstract

Although digital technologies on farms bridge the productivity-sustainability gap, they are not yet widely adopted. Familiarising farmers with digital systems is as important as developing advanced technological platforms: As farmers become accustomed to and feel in control of digital systems, they will find it easier to accept, adapt and keep pace with technological developments. This article describes the design and implementation of a flexible, scalable, easy-to-use and extensible IoT embedded system to control sprinkler irrigation in an outdoor Thuja conifer nursery in an automated mode and under varying weather. Irrigation is controlled by Mamdani Fuzzy Inference Logic based on rainfall prediction and real-time monitoring of the plants. The IoT system provides a control dashboard and offers three operating modes: manual, automated or scheduled (daily, weekly, and monthly). The system is robust in the event of a power outage or loss of connectivity. The code is available on GitHub. Sensors, solenoid valves, Raspberry Pi microcontrollers, fuzzy logic systems, a web interface and a cloud service create a sustainable solution that makes water use more efficient, creates a healthy environment for crops (using just the right amount of water), makes life easier for farmers, and exposes them to the benefits of the IoT.

KEYWORDS

automation, embedded systems, Internet of Things

1 | INTRODUCTION

Digital innovations in agriculture are necessary, not only in underdeveloped countries experiencing food and water scarcity but also in developed countries because of climate changes, wars, natural disasters, migration, pandemics, or other unexpected challenges. For the sustainable development of our society, we need the concepts of circular economy to be ubiquitous, and agriculture is no exception. Advances in artificial intelligence, genetic algorithms, fuzzy rules, computer vision, machine learning, drones, robotics, and IoT systems significantly support digital innovation in agriculture.

Due to climate changes and excessive global warming, water consumption is increasing yearly and has risen considerably in recent years. The agriculture sector uses 70% of total water [1], most of it for irrigation to ensure the growth of crops, but it is used excessively, often uniformly, on the entire cultivated area, without considering the plants' humidity needs. Crops can grow without soil, but without water, it is impossible. Efficient water usage methods in agriculture are needed to avoid over-irrigation and soil erosion, which lead to crop damage. Also, sustainable food sources that could solve world hunger and malnutrition are required. An efficient irrigation process increases food production quality and quantity [2].

With food security becoming an important issue, higher yield and improved food quality need to be assured globally. The agricultural industry is the primary source of food production, and improving the agriculture sector will directly impact the world hunger problem. Industry 4.0 combines IoT, automated systems, wireless sensor networks, big data, analytics, and cloud solutions in agriculture to create an efficient environment that provides better yield, food quality, and quantity [3, 4]. Many implemented automated smart systems prove that using IoT solutions in agriculture to control irrigation does improve water management and distribution, creating a healthy environment for the crops. The agricultural sector needs to replace traditional irrigation methods with efficient, smart, automated, scalable approaches to resolve the wastage of resources.

Collecting crop environment data like soil moisture, air temperature, and humidity is recommended for effective irrigation. The system's decisions and actions are dependent on these collected values. The most used sensor in IoT irrigation systems is the soil moisture sensor which provides the key parameter in an irrigation process [5, 6].

This article presents a research project where the irrigation process is controlled automatically and adaptively under varying weather, by combining rain forecasts and real-time monitoring of crops and terrain obtained by sensors through Mamdani Fuzzy Inference Logic. Modifications were applied to state-of-the-art hardware design in order to make the system more scalable, extensible and flexible and also adapt controller design in order to make the system more easy-to-use by farmers. The system was designed with the aim of improving the interaction with the farmer, supporting them in the transition towards smart agriculture.

The rest of the paper is organised as follows: Section 2 presents related work. Section 3 discusses the methodology applied in developing our work, highlighting the architecture of our IoT system and the considered use cases. Section 4 illustrates the results obtained after the system was used. Section 5 addresses discussion, open problems, and comparisons with other referred systems. Finally, Section 6 highlights the paper's conclusions and suggests future research directions.

2 | RELATED WORK

We now briefly summarise the research literature dedicated to IoT systems in agriculture, articulating the differences with our method. A synthesis of research gaps addressed in our work vis-a-vis existing works is presented in Figure 19 from Discussion section.

H. Benyezza, M. Bouhedda, and S. Rebouh [7] implemented a radio-frequency (RF) communication in a WSN IoT system to gather data about different zones of a tomato greenhouse. Controlling the irrigation with a fuzzy logic controller (FLC) and comparing their methodology with three other irrigation methods, their strategy was more efficient regarding water and energy use. However, there are a few gaps in this work, namely excessive radiation caused by RF for vulnerable persons (pregnant women, children, heart patients with pacemakers etc.) and the considerable risk of being hacked. Furthermore, the irrigation time of each zone is calculated based on only two parameters: temperature and soil moisture. An essential difference between the work referred to above [7] and our solution is that we use a Wireless Local Area Network (WLAN) module to ensure an Internet connection via Wi-Fi. All modules send data to a server that hosts an Message Queue Telemetry Transport (MQTT) broker and handles the communication between system components. As a security measure, we are using authentication to send data to the MQTT broker and planning to encrypt the data sent over the Internet. In addition, we are using four linguistic variables (instead of two) for controlling the irrigation. Besides the soil moisture and air temperature, we also use air humidity and light intensity.

E. A. Abioye et al. [8] developed an IoT-based monitoring system to measure the soil moisture of a mustard leaf greenhouse and also used a weather station to compute the evapotranspiration (ETo) of the plants. These two factors were used for the manual scheduling of the drip irrigation process. After running the system for 35 days, the gathered data were used for implementing predictive models for the dynamic behaviour of the soil, plant, and weather condition with approximately 90% accuracy. The research gap that our work addresses is that in ref. [8], the cultivation experiment on mustard leaf vegetable plants was conducted in a greenhouse with no rain. In our work, we control the irrigation for open-air Thuja conifer nursery considering the outside weather conditions besides soil moisture, temperature, humidity, and light intensity.

N. K. Nawandar and V. R. Satpute [9] implemented a remote-controlled irrigation system that uses MQTT and hypertext transfer protocol (HTTP) protocols to present collected data from the field and inform the farmers about the state of the greenhouse or farm. Their low-cost automated irrigation system proved to be more efficient than traditional methods regarding water usage by creating the irrigation schedule based on neural decisions. Even though ref. [9] is one of the closest approaches to our solution, their claimed 'low-cost' irrigation system is much more expensive than ours. Additionally, the crop details fetched from the user are inputted to estimate the water requirement. In our case, the system is robust and runs automatically, and is capable of resuming execution when the power supply or Internet returns.

J. Jiang et al. [10] implemented a WSN automated system for monitoring the temperature and controlling the fan-circulating process in a plant factory. By maintaining the perfect temperature for the lettuce in the solar, the period of the crops' growth decreases. After deploying the system, the results show a 61%–109% weight increase in the lettuces. Their automatic temperature monitoring system used a commercial product called the Octopus II to implement collectors and sink nodes. In our system, we built the collector and the sink components from scratch, making modifying or adding new sensors or nodes easy and reducing costs. The gateway of Jiang collected the sensed data from the wireless sensor nodes every 10 min, while in our system data are sent via the MQTT protocol to the sink node at configurable time intervals (at the moment, 1 min). In an indoor environment, Boston lettuce seedlings were exposed to artificial lights. They required special growing conditions regarding temperature, humidity, and lights. All these increase the cost and energy of the experiment compared with our Variable Rate Irrigation (VRI) system.

H. Zhang et al. [11] developed a LoRaWAN IoT system for the precision irrigation of fresh-market tomatoes with four different irrigation criteria. Their results show that using the MP60 treatment resulted in a higher yield of the crops. Unfortunately, their IoT irrigation system had an average data loss rate of 5.5%, attributed to the indoor gateway placement and disconnection of each component in the network. In our case, even if the power supply or Internet connection is lost, only the data storage is interrupted. However, the set irrigation part will still work (regardless of collectors and weather). In ref. [11], the irrigation system is manually handled, and the irrigation only starts when the operator turns on the corresponding valves.

Furthermore, the system requires higher voltage batteries for driving the solenoid valve. The solenoid did not respond even if the controller received the signal. The valve finally opened or closed by sending the signal several times. Unlike them, we provide three operation modes: manual, scheduled, or automated.

J. Muangprathub et al. [12] implemented an IoT-based irrigation and monitoring system for homegrown vegetables and lemons, consisting of an embedded device, web application, and mobile application. They used soil moisture, temperature, and humidity data for irrigation and data mining algorithms to find the perfect temperature (29–32°C) and humidity (72%–81%) for the highest productivity. The main difference between our VRI system and [12] is that we use light sensors, weather forecasts, and fuzzy rules to control irrigation. In addition, we have a third irrigation mode, scheduled, where the farmer may configure daily, weekly, and monthly plans by setting the time to start and the duration of an irrigation session.

J. Lloret et al. [13] developed a soil moisture sensor based on two coils with which they built a WSN for monitoring soil moisture. Multiple collecting nodes with four sensors were placed among citrus trees for testing. A precise sensor for soil moisture helps the system's sustainability and accuracy. An essential missing component of their work is that they do not provide a web interface for controlling the irrigation by farmers, which our system does. Also, in ref. [13], only a sensor based on coils is used for measuring soil moisture. Unlike them, we employ four types of sensors to provide a complete image of the environment, weather forecasts, and fuzzy control.

Z. Tsiropoulos et al. [14] developed a low-cost IoT system to irrigate various crops using the FAO56 model to compute water requirements. Their solution was tested in open-air environments with multiple crop types and water sources. It proved to be efficient, low-cost, and user-friendly but that system does not use weather forecasts and fuzzy control for irrigation systems and does not provide a web interface for controlling irrigation by farmers.

Z. Mushtaq et al. [15] designed and simulated a fuzzy controller using MATLAB for automatic land irrigation. They used two linguistic variables as inputs: agricultural land water level categorisation and time (day, afternoon, and night). The controller outputs consist of tube well operation and an electric power source. Unlike them, we use a more significant number of linguistic variables and fuzzy control rules. Most importantly, we did not simulate but physically implement a real, working system.

R. S. Krishnan et al. [16] devised a system for monitoring and controlling plant growth irrigation to improve agriculture productivity. Their fuzzy controller uses only three input variables, and if other variables are added, the number of rules rises, and the set of rules becomes hard to compute. The authors do not specify the geographical positions, the climate conditions, or the type of plants. The difference between their work and ours is that they do not provide a web interface or remote control to start or stop the irrigation. The user must be near the display to see whether the sensor works. Our fuzzy system uses four linguistic variables instead of three in their case. We are using the cascaded mode, in which the rule set is easier to define, allowing farmers to modify the rules by their needs. In our application, rain is determined in real-time. We avoid the cost of a rain sensor because we read, with the help of API, some weather forecasts from the national meteorological agency. In ref. [16], there are only two operating modes (we have also scheduled operating mode). Our developed system provides data analytics directly in the web application, no manual data processing is required, and charts are generated automatically.

M.S. Munir et al. [17] presented an intelligent Smart Watering System for handling the watering process of plants by getting response commands from the gardener (manual intervention of the user). Our VRI system provides three operation modes: manual, automated based on sensors data and weather forecast, or scheduled, where the farmer may configure daily, weekly, and monthly plans by setting the time to start and the duration of an irrigation session. Even if their fuzzy logic controller includes four input variables (same as ours), they implement only two rules instead of 33 as we do. Furthermore, Munir et al. [17] does not provide any information about the fuzzy rules set, and the control system does not use any precipitation prediction (as we do). Even though their Smart Watering System uses spotlighted technologies like blockchain [17], it has a major drawback because it needs a high power consumption for data transfer.

B. Alomar and A. Alazzam [18] presented an irrigation system that controls the water flow from the pump with a rotary switch and reduces the irrigation frequency while increasing the production rate through fuzzy logic. Their fuzzy controller uses only two input variables, soil moisture and outside temperature, and implements only nine rules instead of 33 as we do. The authors do not specify the geographical positions, the climate conditions, and the type of growth. The differences between their work and ours are that they do not provide any web interface or remote control to start or stop the irrigation; our fuzzy system uses four linguistic variables. We collect data in a database and display helpful information for the farmers, which is not achieved in ref. [18].

One approach very close to our solution is the one of Keswani et al. [19]. The main aim of this research is to design a soil moisture content prediction and irrigation control methodology using neural networks and bilinear interpolation to generate soil moisture distribution throughout the farmland.

The valve control commands are processed using fuzzy logic by considering different weather conditions (wind speed, temperature, humidity). Although they use environmental information, they do not use weather forecast data on potential rainfall collected from recognised national agencies (as we do) in the next hour. The fuzzy rules implemented in ref. [19] are 18 instead of our 33 rules. The sensed data collection from the wireless sensor nodes is made every 10 min relatively slowly than us, where the data are sent via the MQTT protocol to the sink node at configurable time intervals (at the moment, 1 min). Also, they do not provide any web interface or remote control to configure the irrigation schedule, like in our case, to help the farmer. Our solution was developed at the request of a Romanian farmer using an already implemented manual irrigation system for an open-air environment Thuja nursery. The system had to be controlled daily, with very few input configurations, turning ON and OFF water valves at specific daily moments. We implemented a configurable irrigation system that can be controlled automatically using Mamdani Fuzzy Inference Logic systems or manually based on a web application dashboard. The irrigation process considers future weather conditions and the real-time state of the Thuja conifers. Through such a solution, we achieve rational but optimal use of water, and we reduce the working time of the farmers.

Human factors are an essential aspect. A key element in designing our system was the ease of use for farmers unfamiliar with technology. The lack of digital knowledge and skills is a barrier to adopting digital technology. Gaining farmers' trust and getting them accustomed to technology is a worthwhile objective supporting the transition to Smart Agriculture [20]. It is not enough to devise algorithms that achieve good predictive performance. Ideally, algorithms should be able to expose, explain and motivate their decisions. The simplicity of use and exposure to the final users of elements such as fuzzy rules is an essential design principle. In addition, the digital lock-in, which emerges in the long term as farmers develop a dependence on software to guide their farming practices, may cause them to lose their ability to perform agriculture effectively without digital support [21]. This means digital systems targeted at farmers ought to be as easy to use and intuitive as possible, and they should leave the farmer the option to exercise manual control and supervision. Our system was designed with these principles in mind.

3 | THE IoT IRRIGATION SYSTEM

The concept of the Internet of Things (Figure 1) dates to the end of the 90s. An IoT device includes sensors and processing capacity (perception layer) connected to one or more modules (network layer) that allow communication through the Internet to deliver application-specific services to the user (application layer). IoT facilitates advanced connectivity of devices, systems, and services that go beyond machine-to-machine (M2M) communications. It covers various protocols, domains, and applications ranging from smart cities to smart agriculture and health. In 2014 there were around 19.7 billion interconnected IoT devices and the estimations are 95.5 billion in 2025 [22], of which more than half will exist in the Industrial Internet of Things. In addition IoT is one of the technologies seen as improving supply chains, making them better suited to withstand catastrophic disruptions [23].

3.1 | The conifer nursery

The Thuja conifer nursery is in an open-air environment in a rural area named Garbova, Alba County, Romania. The climate in Romania is temperate continental, with temperatures ranging from an average low of −5°C in January to an average high of 29°C in August. For our research, we used an area of 50 m^2 as shown in Figure 2. The Thuja plants are placed in P9 pots type containers. A fixed system of a solid set of plastic pipes with nozzles with 3 m sprinkling radius (sprinkler heads) was mounted above the pots. Solenoid valves control each pipe's water flow, leading to six pipes and six valves. Before integrating our solution with the pipe system, the irrigation was controlled by a system that needed to be configured manually to start and stop the irrigation process at hour intervals and for a specific amount of time. This process was not considering the crop-water requirement and weather factors, leading to wasting water and forcing nursery managers to pay more attention to the irrigation process.

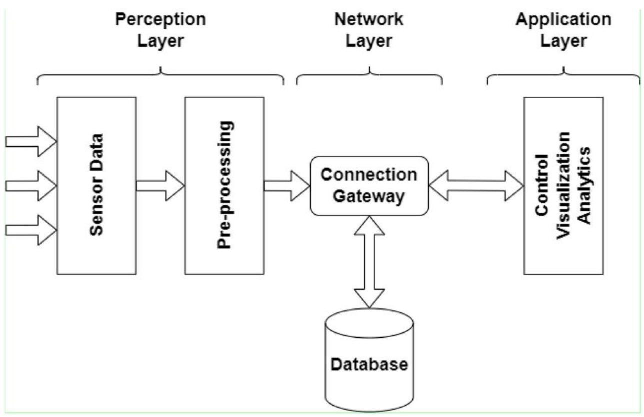

FIGURE 1 Internet of things working layers.

FIGURE 2 Thuja conifers nursery with fixed water pipes system in an open-air environment.

Our goal was to replace the existing system with a smarter, more sustainable, and more efficient IoT irrigation system. Our solution integrates the functionality of the old system in a digitalised manner. It also adds other beneficial use cases, such as: monitoring the state of the nursery, controlling manual irrigation, and controlling and configuring automated irrigation.

3.2 | System model

The proposed IoT system overview is presented in Figure 3. The data-collecting devices from now on referred to as 'collectors', are placed among the Thuja pots. The collectors gather data about soil moisture, air humidity, air temperature, and light intensity. The gathered data are processed and packaged to be sent over the network to a central node. Placed on the house, on the left, is the main controller device of the nursery, from now on, referred to as the 'sink'. The central node receives the data from all collectors and sends commands to control the water valves to ensure the Thuja pots' irrigation. Collected data are uploaded into a database for later representation/analysis in the web application.

For exemplification in this paper, in the field, the collectors were arranged as a matrix with three rows and two columns, using six collectors. A unique ID, a four-digit number, was assigned to each collector. The first two digits represent the line's number, and the last two the column's number. The ID is also written on the protection case to easier identify the right collector in case of failures. An algorithm is implemented in the collector to be identified and added automatically to the sink for receiving and using its data. This way, integrating a new collector into an existing and functional system is easy and fast. No updates or modifications are necessary on the other system components.

3.2.1 | The collector node

Figure 4a shows the hardware structure of the collector node. As the processing unit, we use a Raspberry Pi Zero W version 1 or 2, with a WLAN module that reads data from connected sensors. The data are sent via the MQTT protocol to the sink node at configurable time intervals (at the moment, 1 min).

Each collector uses two capacitive soil moisture sensors, one DHT22 combined air temperature, and humidity sensor, and one photoresistor sensor for light intensity is connected and used at this moment at each collector. All used sensors except DHT22 provide data in analogue format. The General-Purpose Input/Output (GPIO) pins of Raspberry Pi are digital, so an analogue-to-digital converter (ADC) ADS1015 was used. All sensors are connected to ADS1015 except the DHT22 sensor, which is directly connected to a GPIO pin. Data transfer between Raspberry Pi Zero and the ADC module is based on Inter-Integrated Circuit (I2C) serial communication, this being the only way of communicating with the module.

Figure 4b shows the application workflow, which is started automatically after powering on the Raspberry Pi. First, the collector registers to sink; this step will be detailed later. Before sending the data to the sink, read values are checked for redundancy based on a short history stored locally. If a sensor becomes non-functional, the issue is logged in a local file that can be reviewed later, and the data message is sent to the sink without the unavailable reading from the non-operating sensor.

The Raspberry Pi, ADC, and wire connections are placed in 11 × 11 cm IP66 protective cases against humidity, sun, rodents, and continuous exposure to water (Figure 5). Each of the two soil moisture sensors has a 2 m cable to allow flexibility. The DHT22 sensor and the light intensity sensors are taped on the bottom of the case, also protected with 3D printed cases.

3.2.2 | The sink node

The role of the sink node is to receive data from the collectors via the MQTT protocol, upload it to a database and control the valves. Also, the sink node receives real-time valve actions from web application users, configures the irrigation process and system settings in real-time, and executes the automated irrigation process if chosen.

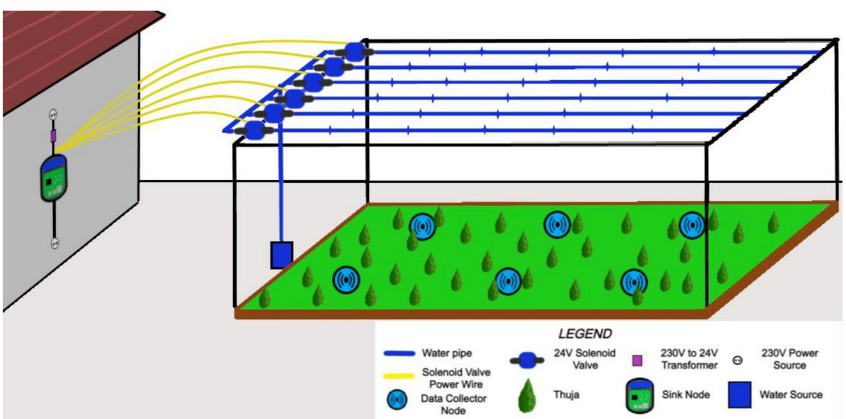

FIGURE 3 Thuja conifers nursery with our IoT system design and fixed water pipes system.

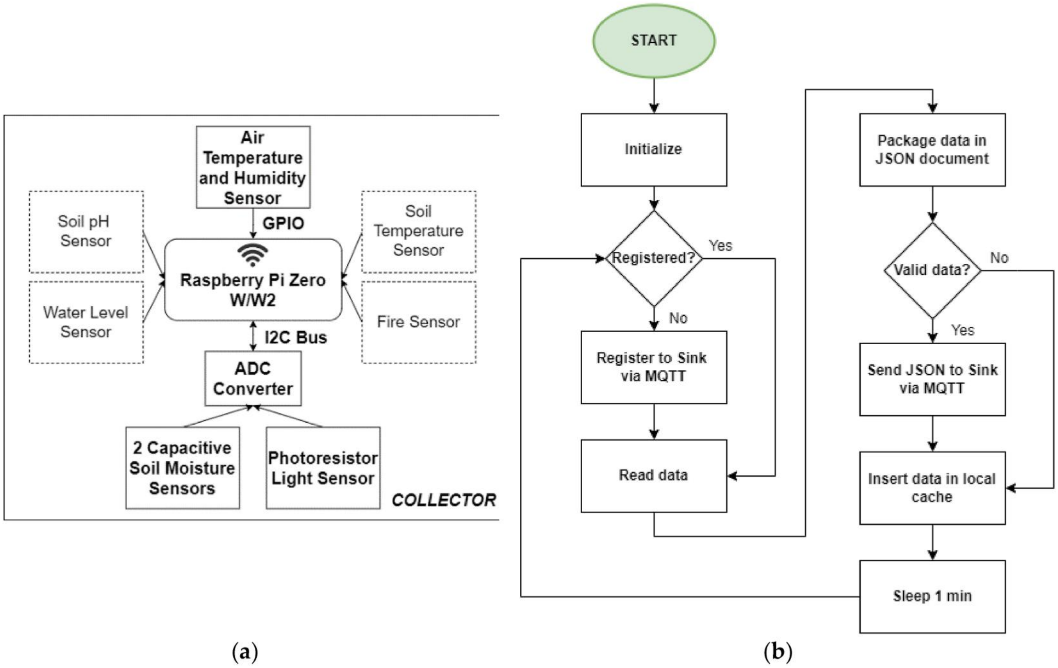

FIGURE 4 (a) Collector node hardware architecture and (b) Collector node application workflow.

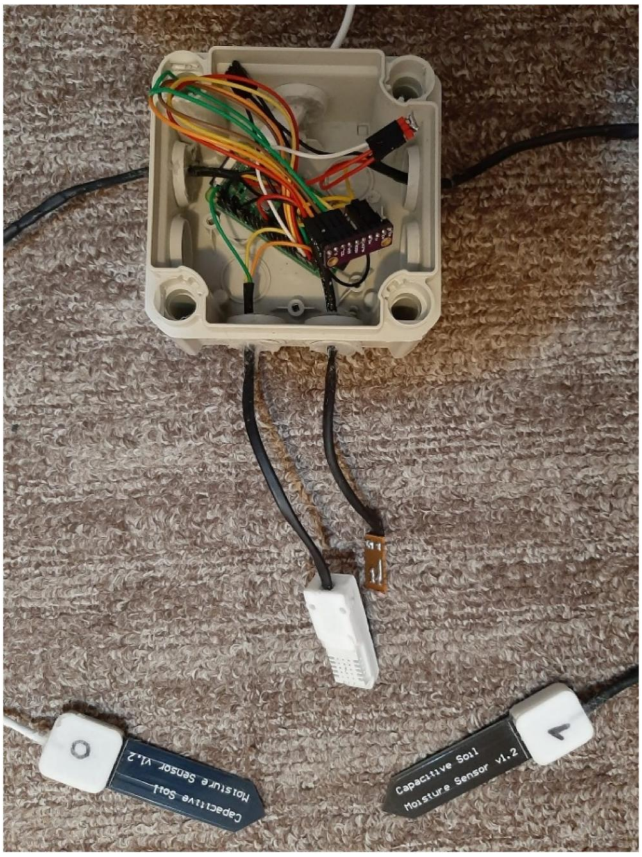

FIGURE 5 Collector node protective case.

Figure 6 shows the structure of the sink node. As the processing unit, we use a Raspberry Pi 4 B+ that controls the 8 Relays Module via its GPIO pins. Each relay corresponds to one 24V solenoid valve. As the system consists of six valves, only six relays are used, and the remaining two relays can be used to control two more valves in case there are added in the future. A transformer was needed because the valves are powered by 24V alternating current.

As application workflow (Figure 7), the application is started automatically after powering on the Raspberry Pi. Initial configurations of the system are stored in an XML file. The application is based on four independent parts:

1. Irrigation process: It is started in the initial mode specified in the configuration file, which can be: manual, scheduled, or automated. Each mode will be detailed later in Section 2.2.4.
2. Collectors' management: This part manages all collectors from the nursery. Each collector must register to send data. This is a security measure that helps the sink to accept data from trusted sources. A record of collector IDs and their registration expiration date is kept. When the expiration date is exceeded, the entry of that collector is removed.
3. MQTT Receiving: A received payload (from collectors or web application) via MQTT is inserted in a thread-safe queue that will be processed later.
4. Payload handling: Each payload added to the queue is categorised by the topic on which it has been received and it will be handled individually after its format is checked. The used events are:
    a. Register event: The payload contains the collector's id. It is added to a record with an expiration date (At the moment, 1 h after registration).
    b. Collector data event: The received data are inserted into the database with its sender collector id.

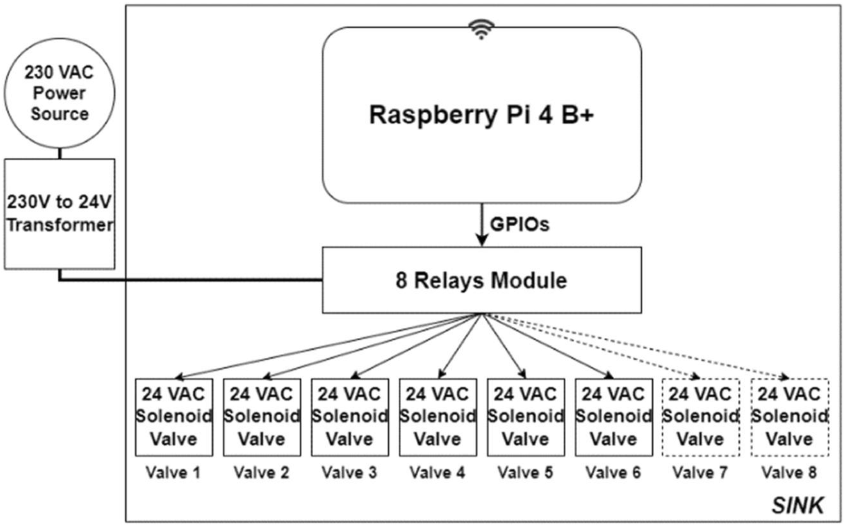

FIGURE 6 Sink node hardware architecture.

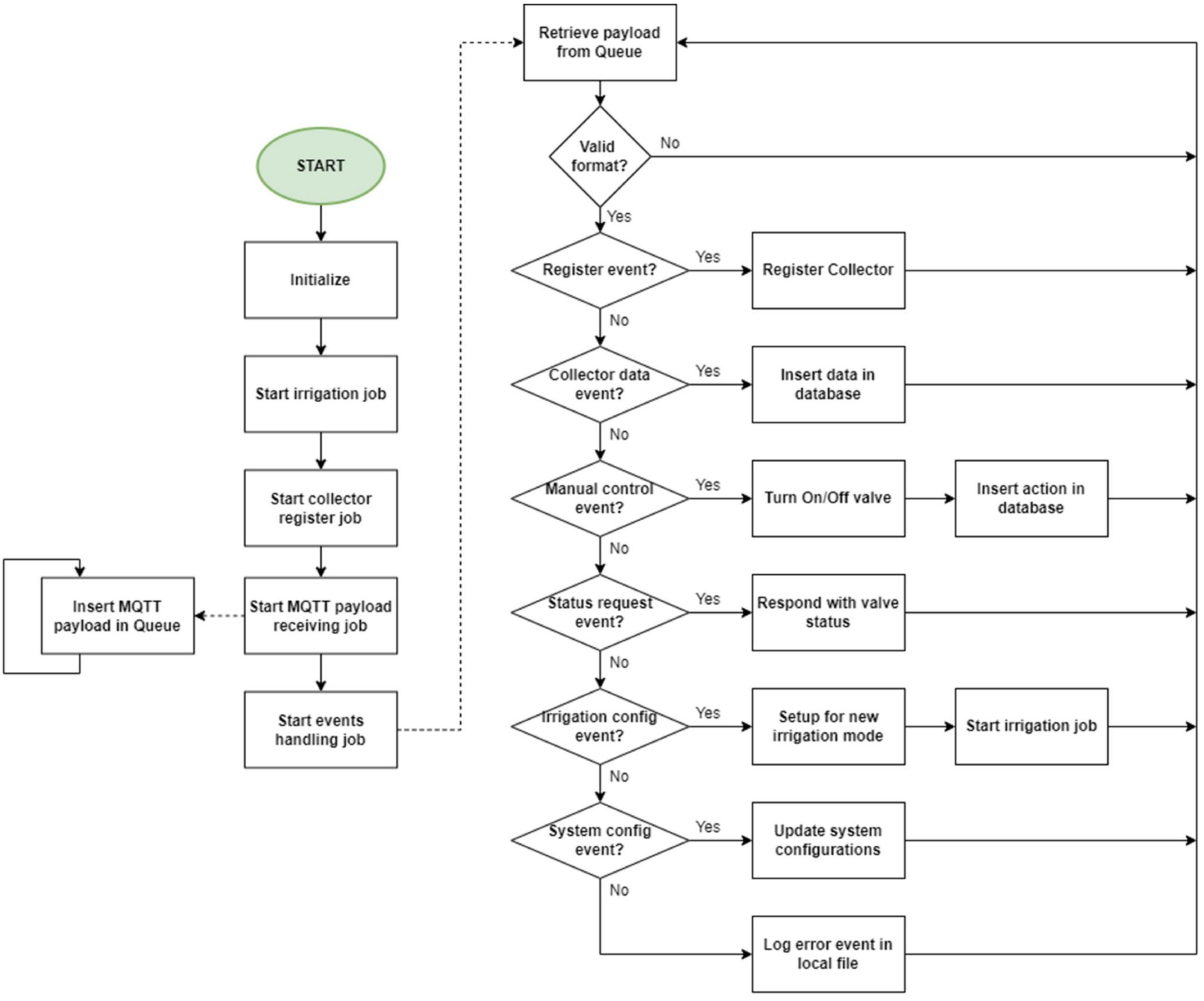

FIGURE 7 Sink node application workflow.

c. Manual control event: The payload contains the wanting state of a valve (ON/OFF) and the user's id who requested it. Web application users may control the valves only in manual irrigation mode.
d. Valve status event: The received payload is empty. In response to the requester, a payload is created and sent with the valves' status. This is requested by the web application to display the valves' state.
e. Irrigation config event: The payload contains which mode the irrigation process will change. The current mode irrigation process is stopped and the new mode is started.
f. System config event: The payload contains new configurations of the system that will be updated in real-time.

3.2.3 | Interconnection

Raspberry Pi 4 B+ and Raspberry Pi Zero W have integrated WLAN modules to ensure internet connection via Wi-Fi. All modules send data to a server that hosts an MQTT Broker and handles the communication between system components (sink, collectors, and the web application). The server hosts a MongoDB database. The database stores sensor data received from the collectors. The database also stores the valves' actions, automated irrigation times, irrigation schedules, and users' credentials provided by the sink.

Figure 8 shows the communication between system components. The MQTT protocol is a lightweight, publish-subscribe network protocol that assures message transfer between devices. The devices do not know the recipient's IP address, only the broker's IP address. The publishers send data to the broker on specific topics, and the broker handles the requests and forwards the data to the subscribers that listen for those particular topics.

Each of the MQTT clients uses level 2 Quality of Service to ensure the data arrives at the destination. All payloads sent over the network respect the JSON format. The payloads are classified by their topic. Each collector sends data (Publish) to the '/collector/data/' topic. The collector is identified based on its ID, which is included in the message. The sink node listens (Subscribe) on the same topic '/collector/data/', processes the received data, and uploads it to the database (on the server). To validate the incoming messages, first, the sink needs to receive collector IDs on the '/collector/registration/' topic. The collectors send their IDs when they start and at a specific time interval (at the moment, 15 min). In this way, only messages from registered collectors are accepted and uploaded to the database. The collectors do not have a direct connection to the database, only the sink does. The sink keeps evidence of the registered collectors and unregisters them at a specific time interval (at the moment, 1 h). In this way, the sink knows which collector did not register again, meaning an error has occurred.

Through relays, the sink node controls the valves (turned ON or OFF). The sink node is subscribed to the '/valves/control/' topic and listens for valve commands. If a command occurs, it processes the message and triggers the relay that controls the specific valve (each relay corresponds to a particular valve).

The valve status needs to be requested on the topic '/valves/status/request/' and the state of the valves is transmitted on the topic '/valves/status/response/'. These topics are used for the web application for real-time visualisation of the valves' status.

The topic '/irrigation/mode/' is used for setting a new mode of irrigation and configuring it. Configurations for the sink are made on the '/sink/config/' topic. For both, forms are

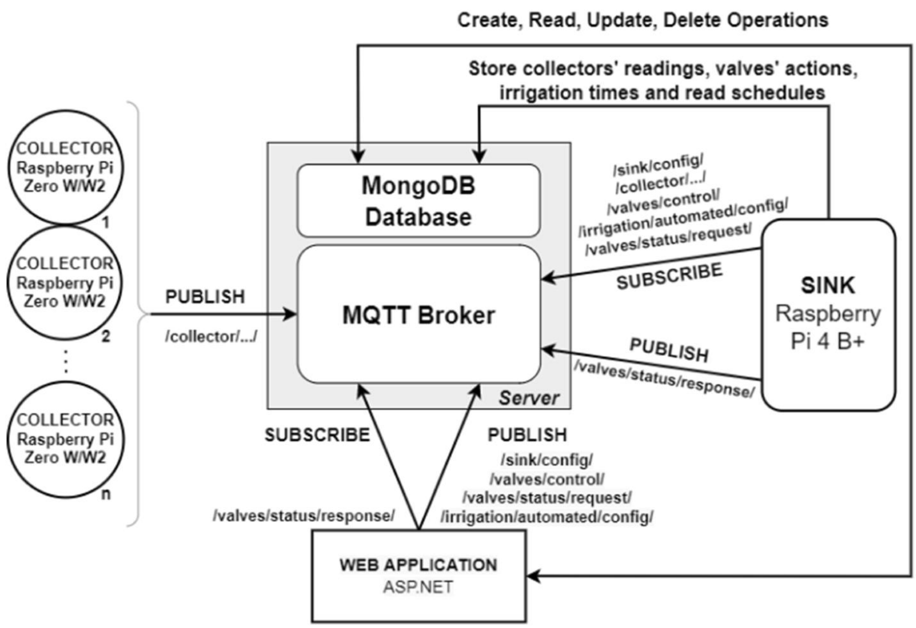

FIGURE 8 Communication between system components.

displayed on the web application, allowing the users to update the configurations in real time.

MongoDB is a NoSQL database that stores entries in key-value types. Collections replace the table format of SQL. Our database uses the following collections: '*users*' which contains web application user information for authentication and contact, '*roles*' which contains web application user roles, '*readings*' which contains collectors' gathered data, and '*valves*' which contains valves' performed actions, '*irrigations*' that contains automated irrigation times and '*schedules*' that contains irrigation plans set by users.

The MQTT Broker and the MongoDB database are secured with user authentication.

3.2.4 | The irrigation process

The irrigation process can be done in three modes: manual, scheduled, or automated.

In **Manual mode**, a web application with a real-time dashboard is implemented where the nursery managers can control each valve individually (turn ON and OFF). Each action will be stored in the database to keep a history of who executed what commands.

In **Scheduled mode**, irrigation schedules can be created and deleted by users from the web application. Daily, weekly, and monthly configuration plans can be established by setting the time to start and the duration of an irrigation session. In this way, we keep the old system's functionality, but we improve the process by digitalising it.

When new plans are added/removed by the user, the schedules are inserted/deleted in the database, and a payload is sent to the sink via MQTT as a notification to update the internal schedules. The sink keeps the schedules in a queue that is checked periodically.

In **Automated mode**, the system automatically calculates and starts the irrigation process, based on the information received from the sensors and future weather conditions. Mamdani Fuzzy Inference Logic systems are used to calculate the time for irrigation based on collected data from the nursery: soil moisture, air temperature, air humidity, and light intensity [24]. Fuzzy Mamdani systems were initially designed to mimic the performance of human operators controlling specific industrial processes. The aim was to summarise the operator's experience into a set of IF-THEN (linguistic) rules that could be used by a machine to control the process automatically.

We chose to use this cascaded design of a Fuzzy (Figure 9) system over a single system with four inputs entering the same FLC. Using three separate FLCs eases the rules definition step of the system. Soil moisture and air temperature are used as principal inputs, based on which an initial irrigation time is computed. In the following two systems, the irrigation time is served as input and it is slightly adjusted based on air humidity and light intensity.

The fuzzification step of a fuzzy logic system translates the numerical values from sensors into linguistic variables through membership functions. Each input has defined three membership functions. Soil moisture, air humidity, and light intensity are represented in percentage with values in [0, 100] interval and air temperature is expressed in Celsius degrees in [−30, 50] interval (Figure 10). The irrigation time has five membership functions, the same for both input and output, and it is represented in minutes with values in [0, 10] Interval (Figure 11).

The rule definition step of a fuzzy logic system consists in creating a set of rules ('IF-THEN' conditionals) that calculate the output values based on relationships (AND/OR) between input values. Since we have three FLCs, three sets of rules for relationships have to be defined. In all cases, we chose AND relationships. With the help and knowledge of the farmers about Thuja conifers' living environment, we came up with the rules presented in Tables 1–3.

There are some important mentions:

- When the soil moisture is Wet, the air temperature is ignored and the output is a None irrigation time (reducing three rules to one).
- When the input irrigation time is None, the air humidity and light intensity are ignored and the output is a None irrigation time (as well as reducing three rules to one).

The number of rules is reduced with the cascaded design model compared to the standard one with four inputs into a single FLC. We have a set of $7 + 13 + 13 = 33$ rules, in comparison with 3 (linguistic variables) ^ 4 (variables) = 81 rules. With this system design, the rules are easy and logical to define, and the performance of the system increases [25].

The inference step of a fuzzy logic system applies the set of rules to the input variables to calculate the output. The Mamdani

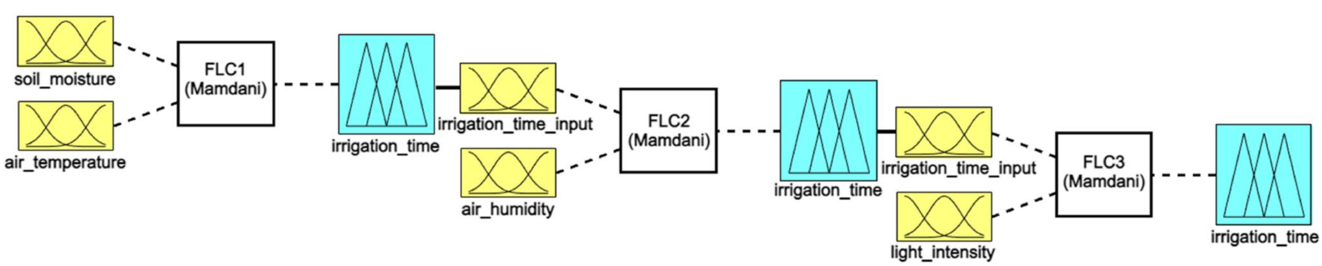

FIGURE 9 Fuzzy system with three fuzzy logic controllers.

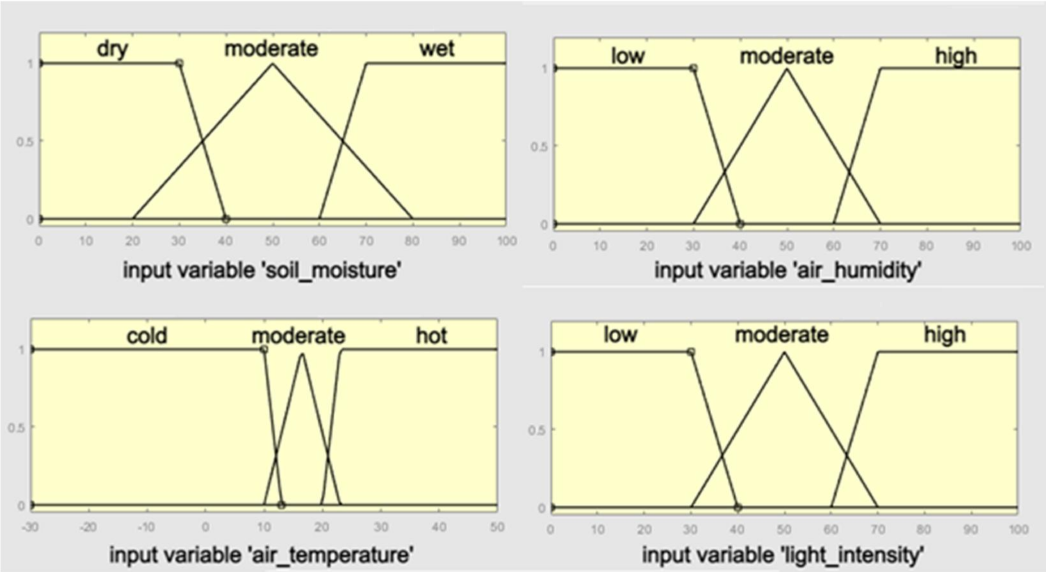

FIGURE 10 Soil moisture, air temperature, air humidity, and light intensity fuzzy logic controller inputs membership functions.

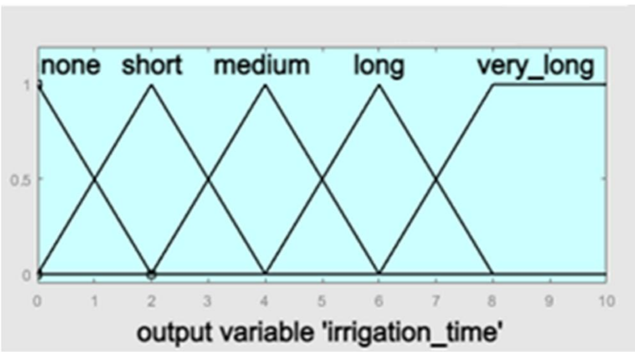

FIGURE 11 Irrigation time fuzzy logic controller input and output membership functions.

Fuzzy Inference System, one of the two basic fuzzy rule types widely used (the other is Takagi–Sugeno–Kang) [26], on grounds of the better interpretation ability, which makes it the preference method in decision support systems.

Once the output value is computed, it is needed to be translated from a linguistic variable into a numerical value. This step is called defuzzification and it is accomplished with the centroid (centre of gravity) defuzzification method.

As workflow, at 15 min time intervals, the last collectors' readings are taken from the database (e.g. the last 20, to correct possible readings errors), and the average value of each input is calculated and served as inputs for the fuzzy logic system. The irrigation time is computed. If it is lower than 2 min, it will wait 15 min, and the process will repeat. After the irrigation time using FLC (Fuzzy Logic Control) was calculated and the value obtained is <2 min, nothing happens. If the irrigation time is >2 min, the system checks the rainfall percentage for the next hour from the AccuWeather API website [27]. If it is below 75%, the irrigation process is started for the calculated time. If the rainfall percentage exceeds the 75% threshold, irrigation is

TABLE 1 Set of rules for fuzzy logic controller 1.

Soil moisture	Air temperature	Irrigation time
Dry	Cold	Long
Dry	Moderate	Very long
Dry	Hot	Very long
Moderate	Cold	Short
Moderate	Moderate	Medium
Moderate	Hot	Medium
Wet	-	None

TABLE 2 Set of rules for fuzzy logic controller 2.

Irrigation time input	Air humidity	Irrigation time
None	-	None
Short	Low	Medium
Short	Moderate	Short
Short	High	None
Medium	Low	Long
Medium	Moderate	Medium
Medium	High	Medium
Long	Low	Very long
Long	Moderate	Long
Long	High	Long
Very long	Low	Very long
Very long	Moderate	Very long
Very long	High	Long

TABLE 3 Set of rules for fuzzy logic controller 3.

Irrigation time input	Light intensity	Irrigation time
None	-	None
Short	Low	Medium
Short	Moderate	Short
Short	High	Short
Medium	Low	Long
Medium	Moderate	Medium
Medium	High	Medium
Long	Low	Long
Long	Moderate	Long
Long	High	Long
Very long	Low	Very long
Very long	Moderate	Very long
Very long	High	Very long

postponed, and the system will check again after 15 min. If irrigation has been postponed four consecutive times, due to the rainfall percentage provided by AccuWeather, the irrigation process will start for the computed irrigation time, ignoring the information from AccuWeather. This approach is done because if it had rained enough outside, the irrigation time obtained due to the sensors would be <2 min and the checks would not be done. After irrigation, the system waits for 1 h to let the soil absorb the water before repeating the process.

The fuzzy system was initially tested in Matlab with the Fuzzy Logic Toolbox and the plots shown in Figure 10 were generated with the same library. The implementation of the fuzzy system in the sink is done with the usage of the Simpful library for Python [28].

3.2.5 | The web application

The web application is based on 7 controllers (Table 4). It respects the Model-View-Controller (MVC) design pattern (Figure 12). It is implemented in the ASP.NET framework. The model represents the form of the data used in the application. It can represent database objects or objects that transfer data from the View to the controller. The View is a user interface, it displays the model data to the user and allows the user to modify it. Through Internet Information Services (IIS) Web Server, the requests from views are routed to the controllers. The controller handles requests, for example, users requesting web pages. The Repository Design Pattern is used for abstraction and centralised data access to the database. Identity Framework is used to manage user accounts and roles.

The web application is created to let the nursery managers interact with their nursery based on a friendly and easy-to-use interface working on different devices. The application allows the users to control the irrigation manually (Figure 13), add

TABLE 4 Controllers used to implement the web application.

Controller	Use case descriptions
Account	Login and Logout (Authentication and Authorisation)
Home	Loading webpage with project presentation. Authentication is not required for view the page. It can be accessed without authorisation, it does not provide any sensitive information
Roles	Create/Read/Update/Delete user roles (e.g., Admin, Viewer)
	Authentication required with Admin account
Users	Create/Read/Update/Delete user accounts
	Authentication required with Admin account
Readings	Read collectors' real time collected data in charts and tables.
	Authentication required with Admin account
Irrigation	Manual: Read valves' real time state Turn on or off valves
	Automated: Read performed automated irrigation times in charts and tables Read information provided by AccuWeather API from the nursery location Applies Fuzzy Rules and send actions to valves accordingly
	Scheduled: Create/Read/Update/Delete scheduled irrigation plans
	Valve Logs: Read history of valve actions performed by users/system
	Configuration: Select irrigation mode (Manual/Scheduled/Automated) Update automated irrigation settings (e.g. Fuzzy Logic System)
	Authentication required with Admin account
System	Provides the addition of a real-time sink configuration
	Authentication required with Admin account

new irrigation schedules (Figure 14), configure the automated irrigation process (Figure 15), visualise real-time data in tables and charts (Figures 16–18), or add a new system setup. To perform all these actions, an admin user account is required to be created by the administrator or by the nursery manager. The viewer user account is allowed only to visualise real-time data.

3.3 | System requirements

A Raspberry Pi 4 B+ microprocessor was used for implementing the sink. Raspberry Pi Zero W versions 1 and 2 were used for implementing the collectors. The used models have a WLAN module for Wi-Fi communication needed for MQTT Broker and MongoDB connections. The software applications for sink and collector were developed in Python 3.7, running on the Raspbian operating system. The web application was built on the ASP.NET framework with. NET CORE 5.0 as the target framework. The server runs Eclipse Mosquitto, an open source MQTT Broker, MongoDB 3.6.5 Community, and IIS 7.5 for ASP.NET web application hosted on a Windows 2008R2 Server machine.

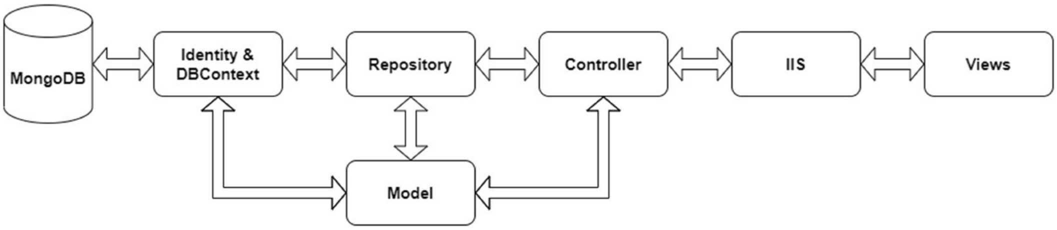

FIGURE 12 Model-view-controller design of the web application.

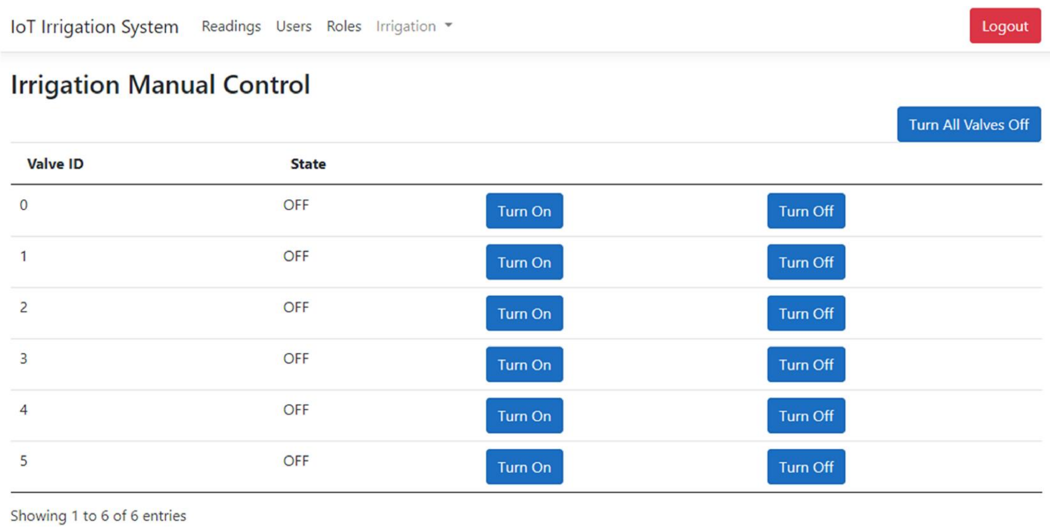

FIGURE 13 Manual irrigation control dashboard accessed from desktop.

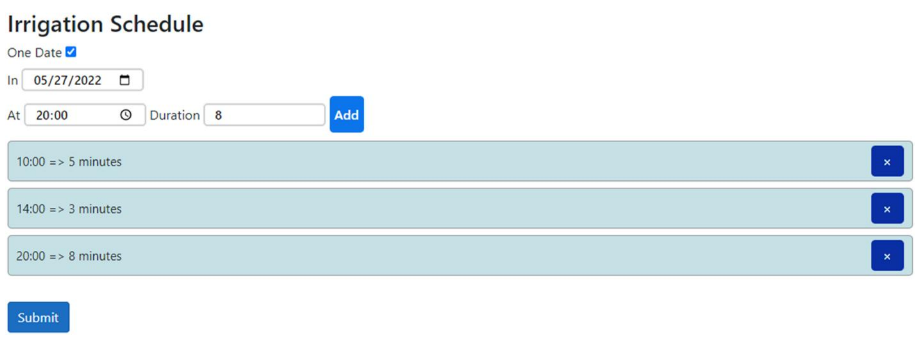

FIGURE 14 Scheduled irrigation dashboard accessed from desktop.

A public GitHub repository for each component was created with the application and system setup steps [29–31].

4 | RESULTS

This section presents the results obtained after implementing and testing the proposed system. Figure 17 shows charts generated in the web application with the collected data during 4 days in June 2022 while running the system in automated mode. Figure 18 shows a chart of irrigation times calculated in the same 4 days. Each data point represents the duration of the irrigation time on which the valves were on. Green bars represent the times when the irrigation was running, and the red bars represent the times when it was not running because the precipitation percent based on AccuWeather was over 75%.

The first chart in Figure 17 presents the collected soil moisture and air humidity in percentage values, the second shows the light intensity also in percentage values, and the third presents the air temperature in Celsius degrees with values in the interval [12, 51]. The soil moisture decreased during daylight when the air humidity was low, and the temperature was high, leading to a higher irrigation time calculated by the system. On the first day, the rain probability was over 75%, and the irrigation did not start until it was under 75%. Even though the rain probability was higher than the threshold, in the end, it did not

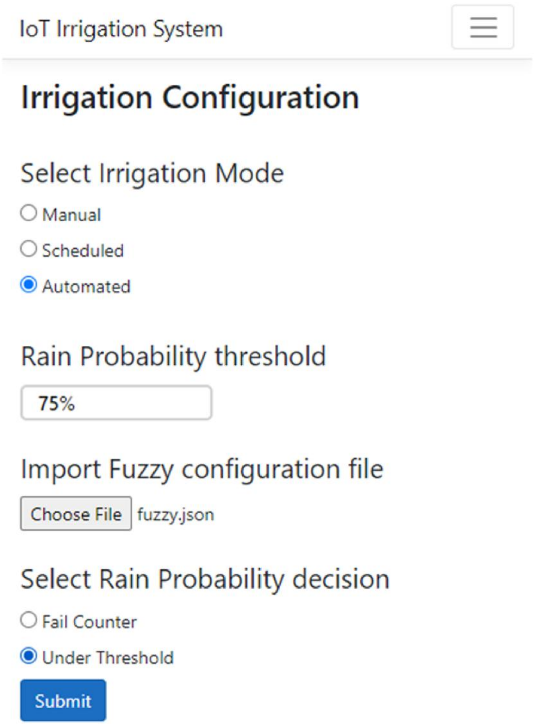

FIGURE 15 Automated irrigation configuration dashboard accessed from a smartphone.

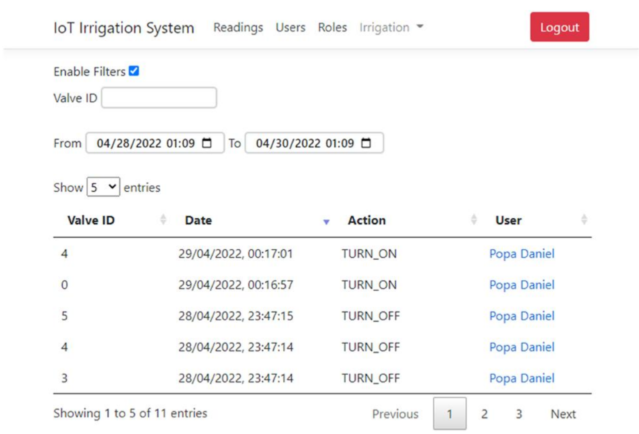

FIGURE 16 List of valves actions logs accessed from a tablet.

rain. During the nighttime, the light intensity and air temperature were low, but the humidity was high, leading to increased soil moisture values and no need for irrigation.

The nursery managers always configured the precedent irrigation system to run six times a day for 4 min each, which means 96 min in total in 4 days. With the automated system, the Thujas were watered for 191.8 min in 4 days, which means two times longer periods of the irrigation process. But, these 4 days were sunny with an average of 28°C in the shade and 51°C in the sun midday, which can be seen in temperature values in the second chart of Figure 17. The precedent system would not ensure the need for water of the Thujas in these conditions, and the farmer needs to intervene to supply more water manually. The volume of used water was not reduced, it was twice higher using our system, but the plants needed this water. The water could be saved when it is raining by not irrigating and when the crops have a high amount of water in their soil.

In a longer evaluation period (>1 month), the benefits of this system can be seen in water saving, but especially in freeing the farmer from the worry of changing the system configurations whenever the weather conditions change.

5 | DISCUSSION

By connecting sensors, water valves, Raspberry Pi microprocessors, fuzzy logic systems, cloud services, and a web application, we implemented an IoT-embedded system (hardware-software) that helps farmers monitor the crops and control the irrigation process in their nursery. Our irrigation system was motivated by the nursery managers who requested a digital replacement solution for their manual system. The proposed system offers a practical and easy-to-use remote-control interface of the nursery via a web application, monitors the environment of the crops in real-time, and offers manual and automated modes for the irrigation process, making the water usage more efficient and allowing farmers to spend less to no time for this job. The cost of hardware components of the proposed system is approximately 30€ per collector and 80€ per sink. Our VRI system has the following advantages:

- Adaptability to weather conditions without farmer intervention every day (especially on rainy or hottest days)
- Fuzzy irrigation control system automatically adapts using weather forecast information
- The architecture of the system allows very easy extension of the irrigation system for larger Thuja crops, with many sensors and water pipes
- Data collecting in a database and applying data science algorithms for predicting suitable temperature, humidity, and soil moisture for optimal future crop growth management.

Currently, the proposed solution faces a few limitations: good Wi-Fi signal coverage is required for good communication, and the server needs to be online.

Compared with mentioned works in the Related Work, our solution can be used for larger-scale nurseries or farms because of our easily extensible architecture. The irrigation process considers near-future weather conditions provided by AccuWeather's open-source API [27] and is controlled automatically and adaptively under varying weather, by combining rain forecasts and real-time monitoring of crops and terrain obtained by sensors through a Mamdani Fuzzy Inference Logic. Protective boxes against rain, sun, and rodents were created for field data collecting devices with 3D printed sensor cases glued with silicone for water protection. Our solution supports three irrigation modes: manual, scheduled, and automated. Instead the authors in refs. [7, 9] used only automated control, and Zhang et al. [11] used only manual control. Muangprathub et al. [12] use both irrigation modes configurable from an

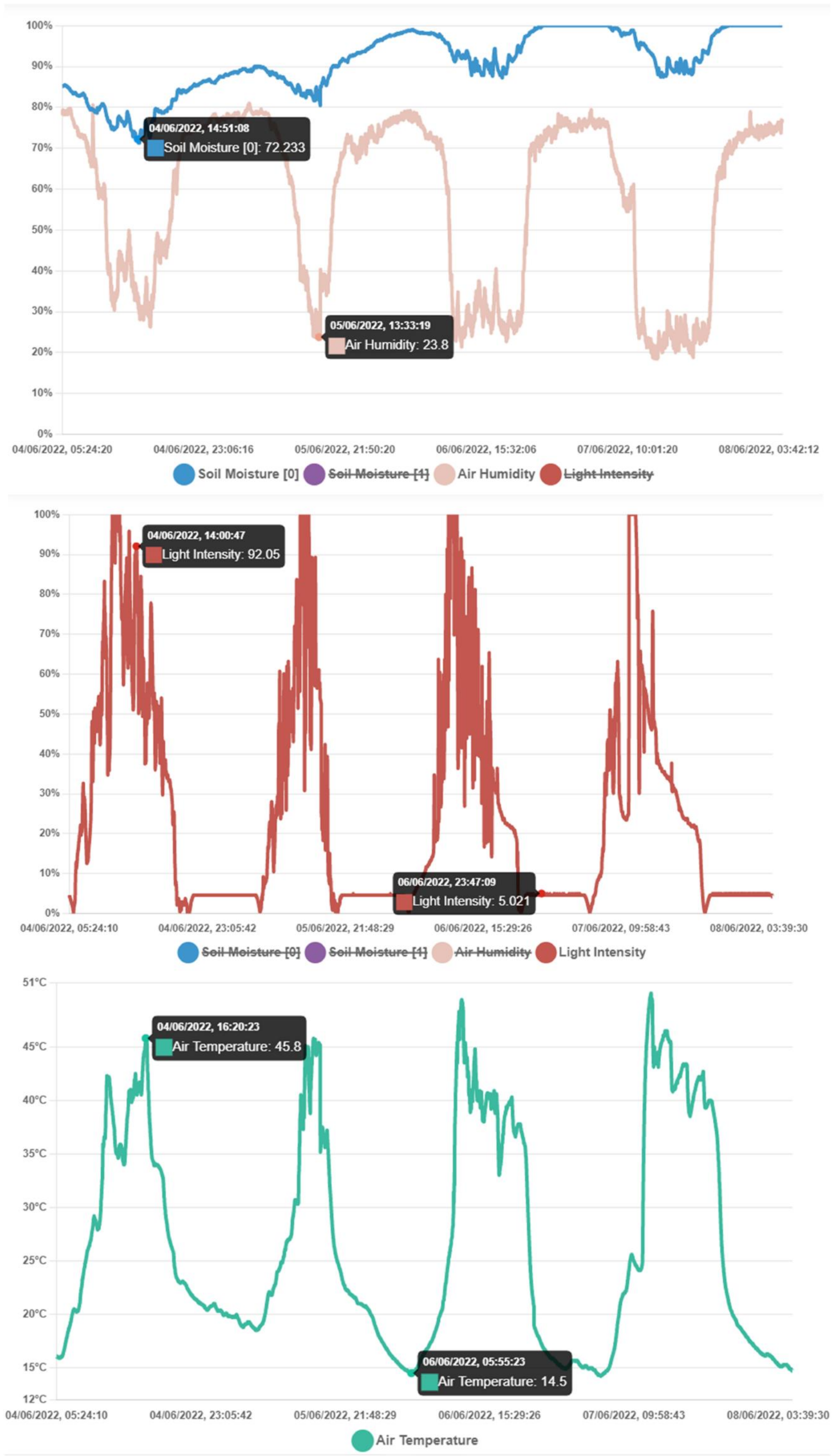

FIGURE 17 Collector readings.

Android app but restricting its applicability to Android smartphone owners. In our work, implementing this functionality in a web application allows users to control irrigation from any device (laptop, desktop, and IOS). Most of the cited papers deal only with certain perspectives (features) of our system. We provide a holistic approach which aggregates many of facilities existing in literature or in some commercial systems. Figure 19 shows a comparison between our proposed method and the previous irrigation solutions. The proposed system offers many additional advantages that can help the user in terms of efficient water usage, smart and friendly control, robustness in case of losing the Internet connection or the power supply, automatic, manual and scheduled remote monitoring and control.

The main contribution of our VRI system does not lie in a specific subsystem: The overall architecture and the

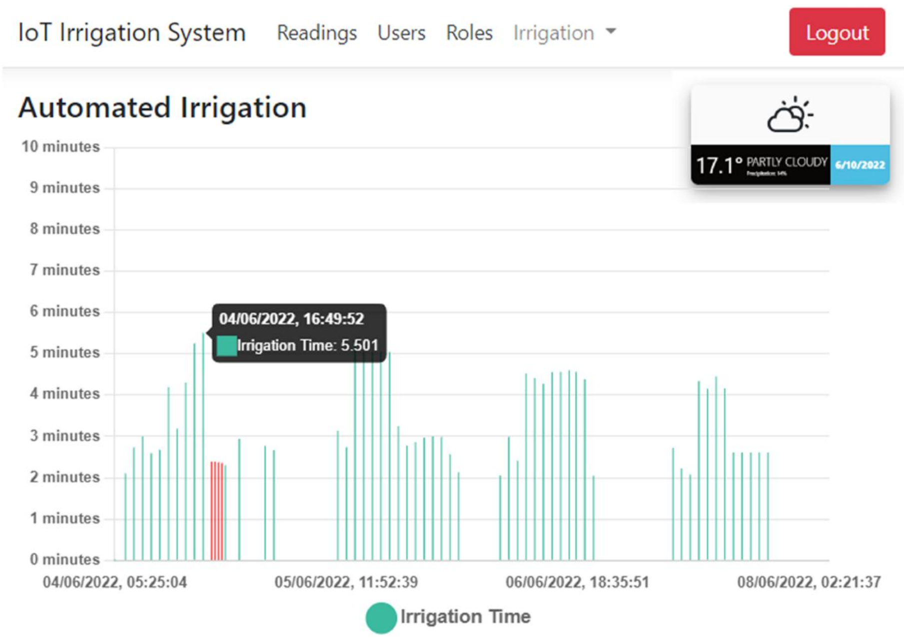

FIGURE 18 Automated irrigation duration times.

Feature / Reference	Communication	Security	Fuzzy: Linguistic Variables	Fuzzy: Rules	Consider Weather forecast?	GUI	Control irrigation	Robustness mechanism	Microcontroller	Open air
[7]	WSN with RF WiFi for cloud	HTTPS User + Password for dashboard	2 (temperature, soil moisture)	25	No	Node-RED Server dashboard	Monitoring	-	Arduino Nano + Rapberry Pi 3	Yes
[8]	-	-	-	-	No	Website dashboard MATLAB simulated model	Monitoring Manual control	-	ESPresso Lite V2 Raspberry Pi 3	No
[9]	WiFI for MQTT	-	-	-	No	Mobile dashboard Website dashboard for configurations	Monitoring NN based decision automated control	-	ESP8266 based controller	Yes
[11]	LoRaWAN	-	-	-	No	AllThingsTalk dashboard	Montoring Manual control	-	Vinduino Board Sentrius™ RG191 (LoRaWAN gateway)	Yes
[12]	WiFi for HTTP	-	-	-	No	Website dashboard Mobile dashboard	Monitoring Manual and data mining based automated control	-	NodeMCU	Yes
[14]	GPRS for cloud	-	-	-	No	-	FAO56 model automated control	Yes	Arduino	Yes
[15]	-	-	2	Not specified	No	-	Matlab simulation	-	-	No
[16]	GPRS RS232 Serial Bus	-	3	27	No	LCD	Manual and fuzzy logic automated control	-	Arduino Uno	No
[17]	WiFi for HTTP	Blockchain	4	2	No	Mobile dasboard	Manual and fuzzy logic automated control	-	Arduino Uno	No
[18]	LR-WPAN	-	2	9	No	-	Fuzzy logic automated control	-	-	No
[19]	Zigbee	-	3 (wind speed, temperature, humidity)	18	No	-	Fuzzy logic automated control	Yes	-	No
Proposed system	WiFi for MQTT and HTTPS	HTTPS User + Password for authentication for website User + Password for authentication for MQTT Broker Register process with predefined ID for each device	4 (temperature, soil moisture, humidity, light intensity)	33 Cascaded mode	Yes	Website dashboard Fully flexible web application for configuring the VRI system & provides charts & statistics	Remote monitoring Manual, fuzzy logic automated and scheduled control	Yes, in case of losing the Internet connection or the power supply, the irrigation will still work!	Raspberry Pi Zero W Raspberry Pi Zero W2 Raspberry Pi 4	Yes

FIGURE 19 Comparison between our proposed variable rate irrigation system with the traditional irrigation strategies.

functionality of our system make it innovative. The implementation combines the most actual and open-source technologies (MQTT, MongoDB, Python, .NET MVC Web Application), so that an efficient, robust, configurable, and sustainable system is created. Also, in the future, TLS/SSL encryption can be added to secure data transmission without risks. The fact that it was tested in a real environment, even in rainy conditions, compared to other prototypes from the literature, is a great plus. More collectors can be added for more extensive nursery coverage, limited by the 4-digit ID representation. The MQTT Broker permits unlimited clients to connect. Therefore it implies no restriction for the number of connected clients, but many clients may cause network latency issues.

In the future, other sensors can be connected to each collector to gather as much data as possible about the plants' living environment. For example, a soil pH sensor to observe different plant behaviours depending on the composition of the soil can also be used. If the selected sensors transmit the data digitally, they can be connected directly to the Raspberry Pi. The sensors will be connected to the ADC if the data are transmitted analogously.

The code of the system is available for free on the GitHub repository [29–31]. Anyone can access it and develop their own system with no other cost than the hardware components. We managed to cover the under-developed parts from other systems and combine all of them into a single working, efficient system. The rain prediction that we used in the fuzzy logic controller is an important factor in watering plants which could save water. As far as we know, we are the first researchers investigating this aspect. Although humidity sensors provide instantaneous readings of local conditions, they only allow a reactive response to weather conditions [32]. By contrast, the use of weather forecasts, based on sophisticated models and large amounts of data, enables a proactive approach that would perhaps be impossible—and certainly not economical—with local resources alone, given the small number of sensors and limited computing capacity. No other systems from the literature use this technique in their implementation.

Our solution helped the farmer who requested to implement this automated irrigation system. Today, the farmer is still using the system and has been able to save time spent on the irrigation process, to focus on other tasks. We consider our VRI system more than a pilot project that works, although any large experiment starts with a prototype. The merit of this work also lies in the entrepreneurial value, namely in educating and familiarising farmers in Romania—and other countries—with digital technological systems. Digital technologies that offer automation of farm operations are seen as key solutions to bridge the gap between productivity and sustainability but have not been widely adopted yet [33]. One of the researchers' conclusions was that farmers over 50 years old, many of them lacking the digital knowledge and skills or higher education to understand how digital technologies can make their jobs easier, would accept digital systems but as easy to use and intuitive as possible [34].

6 | CONCLUSION

This work describes a user-centric digital solution for Thuja farmers. The real-time monitoring of Thujas' environment via a dashboard offers the possibility to present the collected data in a friendly manner. This functionality is available to the nursery manager in an easy-to-use web application, accessible on all digital devices like desktops, laptops, tablets, or smartphones. Other design principles are the system must be reliable and durable.

Learning to use and control IoT systems and adapting to a technology that changes at a hectic pace is very important in the digitalisation of the agricultural sector. Our system has 'plug and play' devices (the collector and sink nodes) that require only a power supply to become functional. In this way, farmers do not need to know and worry about how to configure the devices. By using the web application to control and monitor the irrigation process, farmers are not completely excluded from the digital process but become an active part of it.

The main advantages of our work are, besides that it works in a real environment, that the system is flexible because most of the settings of the three applications are configurable, that it is easy to use, that it is scalable to many areas or types of plants because the fuzzy logic can be configured from the web interface, and that the automated control of the irrigation helps farmers and reduces water waste. Integrating other collecting devices is a simple process, resulting in more extensive coverage of the nursery and improving the accuracy of the automated irrigation process. Storing the collected data in a database allows for analysis and future development in predicting behaviours related to different weather conditions.

For future development, we intend to improve the security of the system, integrate a GSM module to connect isolated nurseries, implement an irrigation backup plan in the event of the server or collector data being unavailable for a long period, and introduce sustainable power sources, for example, solar panels and rechargeable batteries. Our plans also include a new embedded system that constantly captures images and identifies crop growth—and possible diseases—with the help of image processing. Finally, applying machine learning algorithms to the collected data will allow the prediction of appropriate weather conditions and soil moisture for optimal management of future crop growth. To this end, the most appropriate machine learning model is to be selected and tuned, always keeping in mind the meaning of the data, to ensure that the predictions are reasonable. Explainability will also be explored because if the predictions made by the model were also explainable, it would greatly enhance its acceptance.

AUTHOR CONTRIBUTIONS

All authors contributed equally to the research presented in this paper and to the preparation of the final manuscript. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGEMENTS

This work was partially developed in the project MAPIoT (Modernizing Agricultural Practice using Internet of Things), the grant 20-COP-0019, Education, Scholarships, Apprenticeships, and Youth Entrepreneurship Programme (ESAYEP) in Romania financed by the EEA Financial Mechanism 2014–2021, Cooperation Projects in Higher Education Area.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Data openly available in a public repository that does not issue DOIs.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

ORCID

Adrian Florea https://orcid.org/0000-0003-0278-4825
Ugo Fiore https://orcid.org/0000-0003-0509-5662

REFERENCES

1. UNESCO: United Nations, The United Nations World Water Development Report 2022: Groundwater: Making the Invisible Visible. Paris. https://www.unwater.org/publications/un-world-water-development-report-2022/ (2022). Accessed 25 08 2022
2. FAO: The State of Food and Agriculture 2020. Overcoming Water Challenges in Agriculture. Rome. (2020). https://doi.org/10.4060/cb1447en
3. Abbasi, R., Martinez, P., Ahmad, R.: The digitization of agricultural industry – a systematic literature review on agriculture 4.0. Smart Agric. Technol. 2, 100042 (2022). https://doi.org/10.1016/j.atech.2022.100042
4. Di Martino, B., et al.: Internet of Things reference architectures, security and interoperability: a survey. Internet of Things 1, 99–112 (2018). https://doi.org/10.1016/j.iot.2018.08.008
5. Veerachamy, R., et al.: Exploiting IoT and its enabled technologies for irrigation needs in agriculture. Water 14(5), 719 (2022). https://doi.org/10.3390/w14050719
6. Ayaz, M., et al.: Internet-of-Things (IoT)-based smart agriculture: toward making the fields talk. IEEE Access 7, 129551–129583 (2019). https://doi.org/10.1109/access.2019.2932609
7. Benyezza, H., Bouhedda, M., Rebouh, S.: Zoning irrigation smart system based on fuzzy control technology and IoT for water and energy saving. J. Clean. Prod. 302, 127001 (2021). https://doi.org/10.1016/j.jclepro.2021.127001
8. Abioye, E.A., et al.: IoT-based monitoring and data-driven modelling of drip irrigation system for mustard leaf cultivation experiment. Inf. Process. Agric. 8(2), 270–283 (2021). https://doi.org/10.1016/j.inpa.2020.05.004
9. Nawandar, N.K., Satpute, V.R.: IoT based low cost and intelligent module for smart irrigation system. Comput. Electron. Agric. 162, 979–990 (2019). https://doi.org/10.1016/j.compag.2019.05.027
10. Jiang, J.-A., et al.: Toward a higher yield: a wire-less sensor network-based temperature monitoring and fan-circulating system for precision cultivation in plant factories. Precis. Agric. 19(5), 929–956 (2018). https://doi.org/10.1007/s11119-018-9565-6
11. Zhang, H., et al.: LoRaWAN based Internet of Things (IoT) system for precision irrigation in plasticulture fresh-market tomato. Smart Agric. Technol. 2, 100053 (2022). https://doi.org/10.1016/j.atech.2022.100053
12. Muangprathub, J., et al.: IoT and agriculture data analysis for smart farm. Comput. Electron. Agric. 156, 467–474 (2019). https://doi.org/10.1016/j.compag.2018.12.011
13. Lloret, J., et al.: A wireless sensor network deployment for soil moisture monitoring in precision agriculture. Sensors 21(21), 7243 (2021). https://doi.org/10.3390/s21217243
14. Tsiropoulos, Z., et al.: Development of an energy efficient and fully autonomous low-cost IoT system for irrigation scheduling in water-scarce areas using different water sources. Agriculture 12(7), 1044 (2022). https://doi.org/10.3390/agriculture12071044
15. Mushtaq, Z., et al.: Automatic agricultural land irrigation system by fuzzy logic. In: 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), pp. 871–875. IEEE. (2016). https://doi.org/10.1109/ICISCE.2016.190
16. Krishnan, R.S., et al.: Fuzzy logic based smart irrigation system using Internet of Things. J. Clean. Prod. 252, 119902 (2020). https://doi.org/10.1016/j.jclepro.2019.119902
17. Munir, M.S., Bajwa, I.S., Cheema, S.M.: An intelligent and secure smart watering system using fuzzy logic and blockchain. Comput. Electr. Eng. 77, 109–119 (2019). https://doi.org/10.1016/j.compeleceng.2019.05.006
18. Alomar, B., Alazzam, A.: A smart irrigation system using IoT and fuzzy logic controller. In: 2018 Fifth HCT Information Technology Trends (ITT), pp. 175–179. IEEE (2018)
19. Keswani, B., et al.: Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms. Neural Comput. Appl. 31(1), 277–292 (2019). https://doi.org/10.1007/s00521-018-3737-1
20. Omrani, N., et al.: To trust or not to trust? An assessment of trust in AI-based systems: concerns, ethics and contexts. Technol. Forecast. Soc. Change 181, 121763 (2022). https://doi.org/10.1016/j.techfore.2022.121763
21. Rotz, S., et al.: Automated pastures and the digital divide: how agricultural technologies are shaping labour and rural communities. J. Rural Stud. 68, 112–122 (2019). https://doi.org/10.1016/j.jrurstud.2019.01.023
22. Florea, A.: Digital design skills for factories of the future. In: MATEC Web of Conferences, vol. 290, p. 14002 (2019). https://doi.org/10.1051/matecconf/201929014002
23. Moosavi, J., Fathollahi-Fard, A.M., Dulebenets, M.A.: Supply chain disruption during the COVID-19 pandemic: recognizing potential disruption management strategies. Int. J. Disaster Risk Reduc. 75, 102983 (2022). https://doi.org/10.1016/j.ijdrr.2022.102983
24. Mamdani, E.H.: Application of fuzzy logic to approximate reasoning using linguistic synthesis. IEEE Trans. Comput. C-26(12), 1182–1191 (1977). https://doi.org/10.1109/tc.1977.1674779
25. Gellert, A., et al.: Performance and energy optimisation in CPUs through fuzzy knowledge representation. Inf. Sci. 476, 375–391 (2019). https://doi.org/10.1016/j.ins.2018.03.029
26. Ojha, V., Abraham, A., Snášel, V.: Heuristic design of fuzzy inference systems: a review of three decades of research. Eng. Appl. Artif. Intell. 85, 845–864 (2019). https://doi.org/10.1016/j.engappai.2019.08.010
27. AccuWeather API. https://developer.accuweather.com/. Accessed 25 Aug 2022
28. Spolaor, S., et al.: Simpful: a user-friendly Python library for fuzzy logic. Int. J. Comput. Intell. Syst. 13(1), 1687–1698 (2020). https://doi.org/10.2991/ijcis.d.201012.002
29. Popa, D.: IS-IoT-Collector. GitHub repository. https://github.com/pdany1116/is-iot-collector (2022). Accessed 25 Aug 2022
30. Popa, D.: IS-IoT-Sink. GitHub repository. https://github.com/pdany1116/is-iot-sink (2022). Accessed 25 Aug 2022
31. Popa, D.: IS-IoT-Web. GitHub repository. https://github.com/pdany1116/is-iot-web (2022). Accessed 25 Aug 2022 25 08
32. Ghazikhani, A., et al.: A smart post-processing system for forecasting the climate precipitation based on machine learning computations. Sustainability 14(11), 6624 (2022). https://doi.org/10.3390/su14116624
33. Relf-Eckstein, J.E., Ballantyne, A.T., Phillips, P.W.B.: Farming Reimagined: a case study of autonomous farm equipment and creating an innovation opportunity space for broadacre smart farming. NJAS - Wageningen J. Life Sci. 90-91(1), 1–23 (2019). https://doi.org/10.1016/j.njas.2019.100307
34. Lioutas, E.D., Charatsari, C., De Rosa, M.: Digitalization of agriculture: a way to solve the food problem or a trolley dilemma? Technol. Soc. 67, 101744 (2021). https://doi.org/10.1016/j.techsoc.2021.101744

How to cite this article: Florea, A., et al.: Digital farming based on a smart and user-friendly IoT irrigation system: A conifer nursery case study. IET Cyber-Phys. Syst., Theory Appl. 9(2), 150–168 (2024). https://doi.org/10.1049/cps2.12054

Appendix A

In this annex are presented the hardware circuit diagrams representing the electric connections between all components used both in collectors and sink nodes.

Figure A1

Figure A2

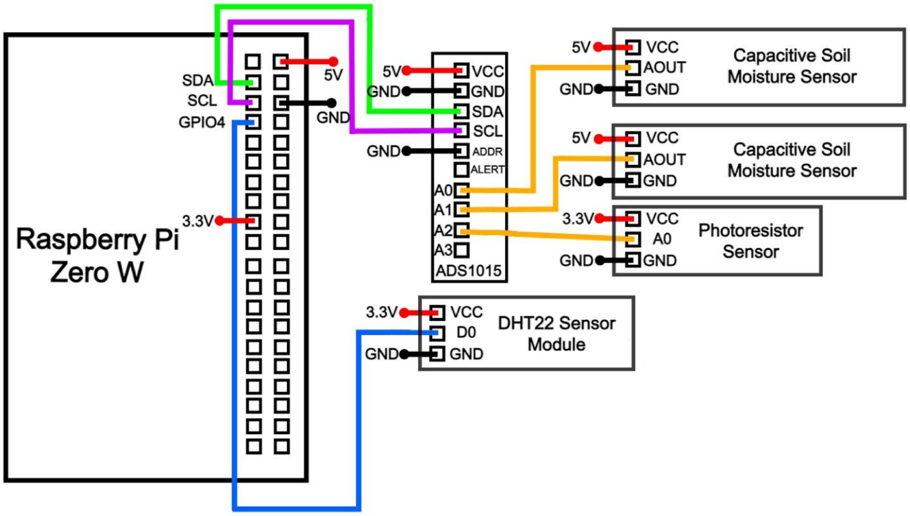

FIGURE A1 Electric connections between all components used in the collector.

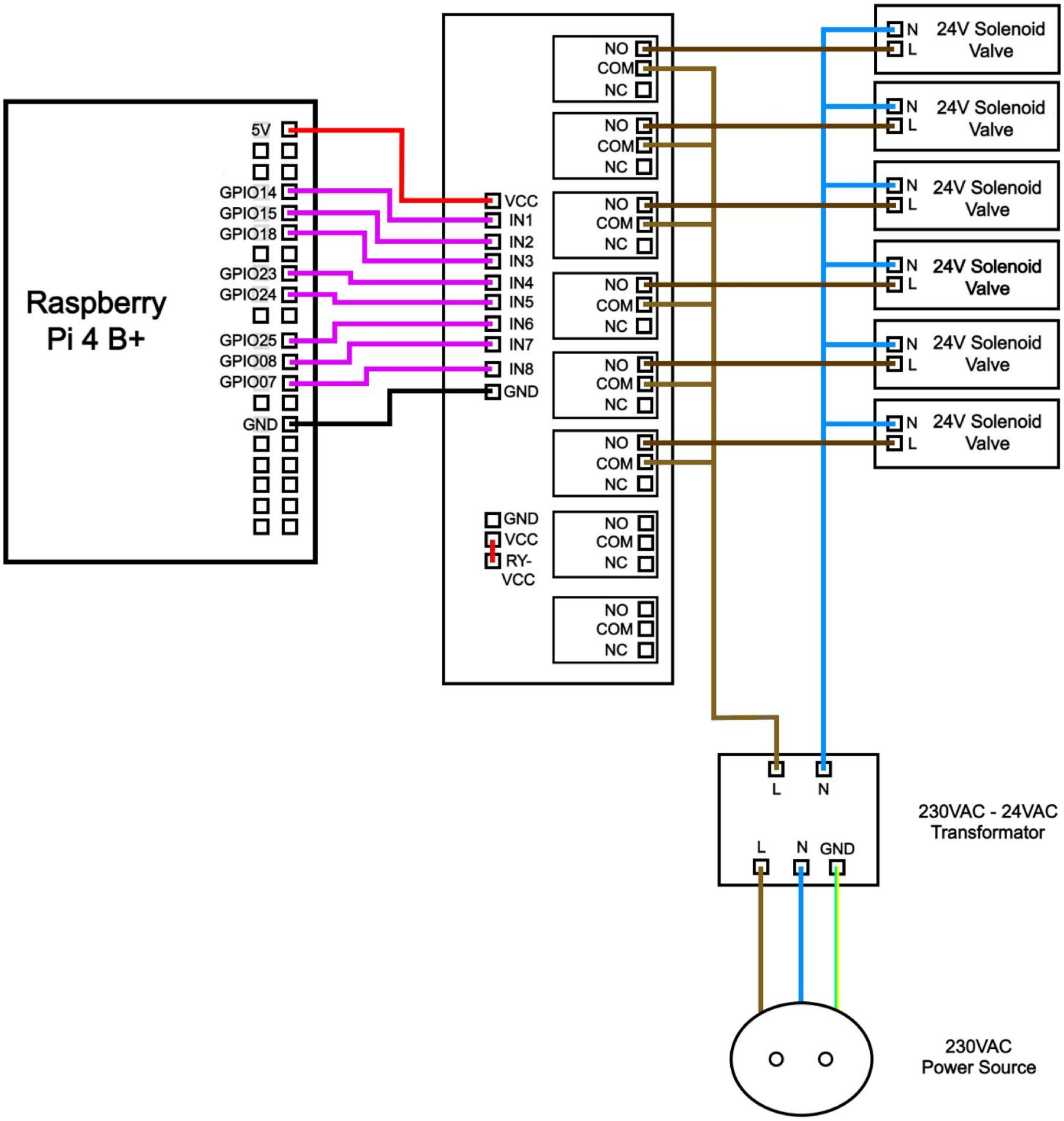

FIGURE A2 Electric connections between all components used in the sink node.