A Climate-Adaptive Smart Irrigation System

Temitope Omotosho Ajayi
Department of Computer Engineering
Obafemi Awolowo University
Ile-Ife, Nigeria
topeajayi@oauife.edu.ng

David Adebimpe Okunade
Department of Agricultural and Environmental Engineering
Obafemi Awolowo University,
Ile-Ife, Nigeria
dokunade@oauife.edu.ng

Rotimi Alagbe Gbadebo
Department of Computer Engineering
Obafemi Awolowo University,
Ile-Ife, Nigeria
oluwarotimi@oauife.edu.ng

Bodunde Odunola Akinyemi
Department of Computer Science and Cybersecurity
Obafemi Awolowo University, Ile-Ife, Nigeria
bakinyemi@oauife.edu.ng

Ganiyu Adesola Aderounmu
Department of Computer Engineering
Obafemi Awolowo University, Ile-Ife, Nigeria
gaderoun@oauife.edu.ng

***Abstract*— Adopting Internet of Things (IoT) technology concepts in developing water and energy conservation techniques for irrigation activities has become imperative in developing countries. With IoT and cloud concepts, a climate-adaptive irrigation system was designed to monitor and present the current condition of a farm using low-cost devices. Thus, this paper provides a smart irrigation system with climate adaptation using decision models to process the data received from sensors (humidity, temperature, soil moisture, and geolocation) put in place on the farm. These data were transmitted to the Smart Irrigation Management software and processed using a rule-based operation that was trained earlier. The processed data were saved in the Smart Irrigation Management software database, and a decision on whether or not to irrigate was made based on the current events on the farmland. The result showed that the system predicted irrigation events on the farmland with 97% accuracy.**

Keywords— Internet of Things; decision model; climate-adaptive; irrigation prediction accuracy; smart irrigation management software

I. INTRODUCTION

Agriculture in Nigeria is characterized by low productivity and limited adoption of technology, resulting in low returns on investment. These restrictions on agricultural activities have hampered agricultural development and sustainability, dragging on Nigeria's economic growth. Modern technological innovations, such as smart irrigation, are required to accelerate the necessary quantum leap toward long-term agricultural growth and development [1]. A major problem in combating food insecurity is overdependence on rainfed agriculture, which makes farming vulnerable to rainfall unpredictability driven by climate change [2, 3]. Droughts are linked to rainfall unpredictability driven by climate change [4]. Droughts can be overcome, and unpredictable rainfall compensated for, through irrigation. Because irrigated land accounts for about 5% of total agricultural land, the irrigation potential could be enormous [2]. Knowing the exact amount of water required for various crops in a region and being able to provide it aids in the planning, scheduling, designing, and management of irrigation systems [5].

Irrigation is generally a time-consuming and labor-intensive activity. Because of the use of technology, irrigation processes that were previously carried out manually have been reduced. Few existing technology-enhanced irrigation systems incorporate climatic conditions, but none have used GPS to address uneven rainfall distribution in specific locations, which leads to substantial water waste, declining water quality, and inefficient energy use for pumping, potentially causing soil water to be excessive relative to plant needs.

Therefore, there is a need for an efficient irrigation system that is adaptable to climatic conditions and location-specific, preventing poor water and electrical energy management, reducing substantial water wastage, mitigating declining water quality that could decrease output, and avoiding inadequate energy use. In this work, a flexible, energy-efficient, and inexpensive smart irrigation system that can be used in any location and adapts to most climatic variations is being developed.

This article is structured as follows: Section 1.0 provides an introduction and background on smart irrigation systems; Section 2.0 reviews existing works on techniques and innovations in smart irrigation systems. Section 3.0 discusses the methods employed in conducting the research; Section 4.0 presents the results and their discussion; finally, Section 5 concludes the research.

II. LITERATURE REVIEW

An automated drip irrigation system that delivers the precise amount of water needed based on soil moisture was developed by [6]. The system collected data from temperature, water flow, and soil moisture sensors. The cloud received the data collected wirelessly. Closed-loop control, in which the microcontroller opens and closes the solenoid valve based on the mean value of soil moisture the plants require, and open-loop control, in which the microcontroller is remotely triggered by the farmer, premised on particular cloud-based data values, are the two

mechanisms of operation for the system. Uddin *et al.* [7] developed a digital solar-powered irrigation system that collects water-level data from croplands and updates the microcontroller and the farmer. Even from afar, the farmer can use a mobile phone to turn the motor on or off based on the water level. When the water level in the cropland nears a dangerous level, however, the motor automatically starts to maintain the proper water level.

A Wireless Sensor Network (WSN) and a General Packet Radio Service (GPRS) modem were deployed to create an automated irrigation system in [8]. The system decentralised wireless connection for soil water content and temperature sensors in the plant's root zone. Data from sensors is processed by the gateway device, which also activates the actuators and transmits data to the web application. To regulate the amount of water, an algorithm was implemented with sensor thresholds encoded into a microcontroller-based gateway. A self-reliant irrigation system was developed by Roy and Ansari [5] that periodically meets certain requirements to irrigate plants. The system used temperature, total radiation, and total wind speed, which were measured by the Programmable Logic Controllers (PLCs), resulting in a reasonable reduction in irrigation water consumption.

Malge and Bhole [9] developed an economical, remotely controlled, simple Embedded System Device (ESD) to manage an irrigation cycle. The PIC18F4550 microcontroller was connected to the GSM module to act as the brain, while temperature and rain sensors served as the eyes of the system. The proposed irrigation procedure starts with the PIC18F4550 energizing the water pump whenever the ESD's eyes detect all parameters within a safe range. ESD on the PIC18F4550 regularly provides the farmer with feedback via SMS. The use of Raspberry Pi and Arduino to create an intelligent drip irrigation system was proposed by Agrawal and Singhal [10]. The system used Arduino microcontrollers and the ZigBee protocol to turn the Raspberry Pi on or off. The Zigbee Star topology is the foundation for communication between the Raspberry Pi and the system. Multiple routers are end systems, with the Raspberry Pi as a key coordinator. The intelligent drip irrigation system is advantageous because it automates and regulates watering without human intervention.

Yadav *et al.* [11] developed a solar-powered plant irrigation system with multiple sensors. The only source of energy for the system was solar power. Cropland was equipped with sensors that continuously monitored water levels and provided the farmer with information. The farmer can use his mobile phone to signal the motor from afar. Once the field's water level reaches a predetermined threshold, the water pump activates automatically, ensuring an appropriate water level in the field. As a result of this system, even if the farmer spends less time on the farm, he can effectively perform his agricultural activities from a remote area. For outdoor agricultural sites, an Arduino-based watering and roofing system was developed in [12]. Sensors provide physical factors, such as moisture, to the system, which is primarily data. The system is based on the Kalman filtering concept, which removes the noise of one sensor to the other to obtain more precise values. When determining when to begin watering, the system employs a decision tree model. A mobile application was created to allow users to receive sensor field data and regulate the application of water and roofing systems from their phones. The system collected sensor data and compared it with weather forecasts using a decision tree model. A Node.js library was used for computational decision-making and integrates machine learning libraries.

An effective, user-friendly, Arduino-based autonomous irrigation system that can be controlled via a smartphone was developed by Arvindan and Keerthika [13]. The HC-05 module, which was connected to an Arduino Uno processor, gathered data from a soil moisture sensor and wirelessly transmitted it to an Android smartphone. The Android smartphone's remote control was used to turn the system on and off. Farmers were able to better control their irrigation time with this system. E-Krisi, a smart irrigation system with IoT and cloud-based technologies, was developed by [14]. The system provided plants with a smart irrigation schedule that maintained adequate soil moisture in the field to improve crop development while reducing waste. Real-time data, such as soil water content and rainfall forecasts for a particular sector, are needed as inputs. It assists farmers in making decisions based on the set thresholds, such as water allocation time and length on the farm.

Kissoon *et al.* [15] launched a smart irrigation and monitoring scheme that employs Azure Machine Learning to analyze farm sensor data and weather forecasts to inform farm owners about irrigation timing. An Arduino development board and sensors measuring temperature, air humidity, and soil moisture were used in the setup. The data collected was utilized to monitor soil water content and air quality. The farmer received information on when to irrigate via a web or a mobile app after these data were processed by a machine learning procedure on the Microsoft Azure cloud platform API. Information on the possibility of rain and if irrigation is necessary was provided to the farmer by the machine learning processing results. As a result, the farmer can remotely turn on the water taps by using a web or a mobile app. Making a well-informed decision when initiating irrigation can save the farmer time and lead to improved water management.

In this study, a climate-adaptive irrigation system that employs decision models to efficiently use water and energy and minimize labor costs on farmland was proposed.

III. METHODOLOGY

A climate-adaptive smart irrigation system comprised of sensors and farm geolocation data is proposed in this paper. The system modules that are adaptive to climatic conditions were identified. Proteus, the Arduino IDE, and the espressif SDK were used in the design of the framework module that was adopted. Then, using data from the designed modules, the rule-based method was used to develop the smart irrigation algorithm. The smart irrigation management (SIM) software in the cloud received data from the farm's sensors, where it was processed using a rule-based operation and a combination of algorithms. The processed data was logged in the database SIM, so the farmer could get notified via the web app when to irrigate based on the event on the farmland.

A. System Architecture

Fig. 1 depicts the system architecture used in this study. The architecture's concepts and formulations are described as follows:

1) Sensor Module

The sensor unit is a combination of 10HS Capacitance and DHT11 sensors. The 10HS capacitance sensor measures the volume of the soil water, while DHT11 measures the atmospheric temperature and relative air humidity. Data from these sensors obtained in real-time were transmitted serially to the Arduino UNO microcontroller**.**

2) The Display Unit

The 16x2 LCD unit conveys the current status of the control unit (CU)

3) The Control Unit

The CU received data from sensor and geolocation detection units and processes it to make appropriate irrigation decisions. This unit processes data from the various sensing units, which include the Arduino UNO and NodeMCU with ESP8266 Wi-Fi module. The control unit's two subunits communicate using a set of AT commands known as Universal Asynchronous Receiver/Transmitter (UART). The NodeMCU's ESP8266 Wi-Fi module acts as a base station as well as a gateway to the Internet, allowing users to access the Open Weather API and Smart Irrigation Management Systems (SIMS).

4) The Power Supply Unit

Using an LM2596 switch-mode regulator that supplies approximately 3 amps of current, the power supply unit's purpose was to reduce the input DC voltage from 12V to 5V. All design components use this voltage, except the NodeMCU, which uses 3.3V and a current of 0.8A after passing the input voltage through a linear voltage regulator (LM1115), which changes the 220V, 50Hz energy source to 5V and 3.3V, respectively. The conversion is achieved by increasing the voltage to 12V (AC) using a step-down transformer (12V, 2.5A) and then converting the voltage to 12V (DC) using a full-wave rectifier. Lastly, to power the control unit and other hardware, the 5V and 3.3V voltage levels were routed through two separate voltage regulators (LM2596 switching regulator and LM1115 linear regulator).

5) Geo-location Detection Unit

The GPS (GY-NEO6M V2) module is used to determine the geographic location of farmland by using signals sent from space satellites and ground stations on Earth to provide coordinates (longitude and latitude). The reading from this unit activates the climatic data for the geographic location to make an irrigation decision.

6) Irrigation Unit

The relay and the water pumping system make up the irrigation unit. The irrigation events activate the relay, which turns on or off the pumping machine**.**

7) Smart Irrigation Management System Unit

SIMS is a back-end web application that notifies users of the system's current irrigation decision. SIMS uses the JavaScript Object Notation (JSON) data structure to communicate with the Control Unit via the internet gateway. All of the events on the farmland are stored in the SIMS, and all of the logs for these events are accessible to all authorised users. SIMS authenticates users via the user interface, and decoded output from the Arduino UNO and NodeMCU microcontrollers is obtained and forwarded to SIMS in the cloud via an Application Program Interface (API).

In addition, the SIMS received weather information from the Open Weather API, which was combined with an irrigation event to notify the user of the present irrigation decision by displaying sensor readings, threshold readings, and rainfall predictions. A specially designed API that uses the JSON standard for communication was queried to achieve interaction between the SIMS and the irrigation system. First Come, First Served (FCFS) scheduling is used to schedule all queries from different units based on their occurrences in the smart irrigation algorithm. The Open Weather API, GPS module, and Arduino UNO sent JSON data strings format to NodeMCU. The words in the strings were divided into distinct substrings for decision-making by using the separator "," during parsing.

8) The User Interface Unit

This unit's user interface is for the Smart Irrigation System. The front-end web interface used by the user interface unit to communicate with the system is depicted in Fig. 2. There is a security precaution which demands valid authentication to keep unauthorised users from accessing the SIMS. Once the user is authorised, the user chooses the operating mode, either manual or automatic. The user has three options: go back to the home screen, manually operate the pump, or exit the system. In automatic mode, the farm location's longitude and latitude readings are shown. The user is notified of the rainfall forecast, any potential irrigation event decisions, and the current sensor data.

Fig. 1 illustrates how users are authenticated by the SIMS using the Smart Irrigation System's user interface. Using an Application Programming Interface (API), the decoded output is sent to the cloud-based SIMS from the Arduino UNO and NodeMCU microcontrollers in the control unit. Furthermore, the Open Weather API provided meteorological data to the SIMS. The user is informed of the current irrigation choice by the SIMS, using the smart irrigation algorithm’s sequence, by displaying sensor readings, threshold readings, and rainfall predictions. When all irrigation events are met, irrigation occurs; if not, it does not occur. Through queries to a specially designed API that adheres to the JSON standard, the SIMS and the smart irrigation system exchange data. Based on their occurrence in the smart irrigation algorithm, all queries from various units are scheduled using a First-Come, First-Served (FCFS) mechanism. The separator "," was used to split the words in the strings into distinct substrings that were utilised for decision- making when NodeMCU received JSON data in the form of strings from the Open Weather API, GPS module, and Arduino UNO.

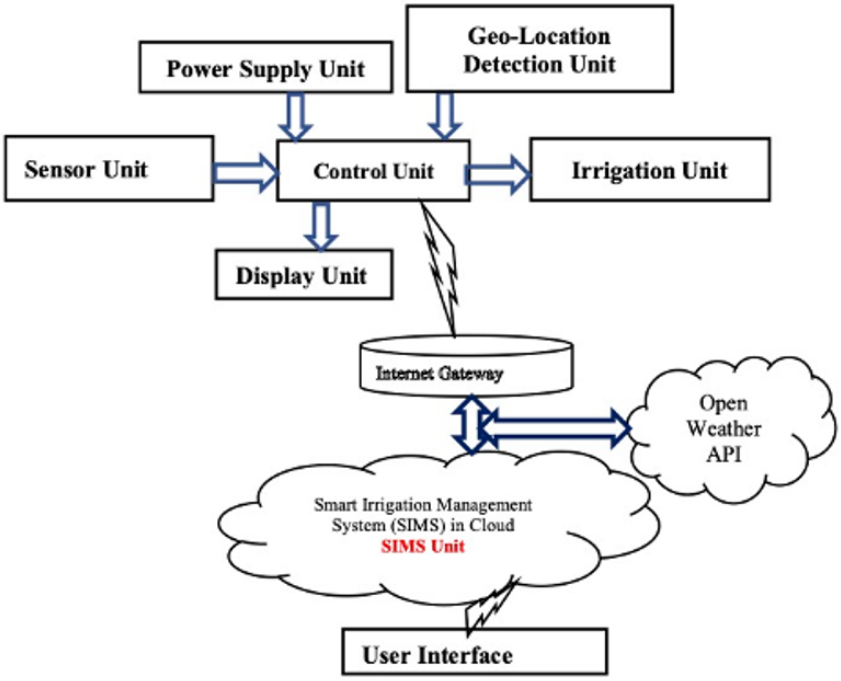


Fig. 1. Proposed Smart Irrigation System Architecture



Fig. 2. Login interface to the SIMS

A. Smart Irrigation Algorithm

The smart irrigation algorithm is implemented through a flowchart shown in Fig. 3, which illustrates the intricacies of the design. The first step is to choose the operating mode of the system (manual or automatic). Using an SMS short code, the water pumping system can be manually turned on or off. It stays in this mode until the automatic mode is chosen. In addition, the GPS module was utilised to pinpoint the position of the farm and establish the temperature, humidity, and soil moisture sensors' threshold parameter values. Additional checks are made for soil moisture level; if an error is detected, the farmer The operation is postponed for 12 hours if there is sufficient water in the soil before the GPS module locates the farm area and resumes. Conversely, the humidity and temperature would be measured if there was no likelihood of rain. Before assessing whether the soil is adequately hydrated, the water pump is turned on for an hour to irrigate the soil if the current temperature is higher than the *ThV* and the humidity reading is lower than the *ThV*.The irrigation process continued for an additional hour provided the soil is not sufficiently hydrated before the soil is examined once again. The operation is paused for 12 hours when there is sufficient water in the soil, at which point the GPS module is used to determine the location of the farmland and the process is resumed.

receives an SMS alert and the procedure is stopped. The system used the Open Weather API to retrieve the weather prediction for the farm location if the moisture sensor value is less than or equal to the threshold value. If not, the procedure stops for one hour before repeating the check. The system checks for the potential of rain; if the weather forecast is incorrect, the farmer receives an SMS and the operation is stopped. When there is a likelihood of rain, the procedure is delayed for three hours before the soil moisture level is remeasured. If the current soil moisture value is less than *ThV*, the irrigation system will be turned on for one hour, and various checks will be made to determine whether to keep on irrigating or stop.

```
1                    Start
2                    Select the operational mode for pumping
3                    IF (pumping mode is manual),
4                                        produce SMS prompt to signify manual pump control
5                                        go back to line 2
6                    ELSE
7                     Pumping mode is Automatic
8                     Check for Location details
9                          Check soil moisture sensor value
10                    IF an error exists,
11                                       send SMS error message
12                   STOP
13      ELSE
14       Check IF soil moisture is less than soil moisture ThV
15       IF (soil moisture is greater than soil moisture ThV)
16          delay 1hour
17                                                          go back to line 14
18       ELSE
19          Check OpenWeatherMap API
20       IF connection error exists
21            send SMS error message
22                  STOP
23       ELSE
24           Check for rainfall prediction
25       IF Rain will fall
26              delay for 3 hours
27              Check soil moisture level again
28        IF (soil moisture level is >= the soil moisture ThV)
29              delay for 12 hours
30              go back to line 8
31        ELSE
32              switch on water pump
33              delay for one hour
34              check soil moisture value
35        IF (soil moisture level is less than soil moisture ThV)
36                 go back to line 32
37        ELSE
38                 Switch off water pump
39                 delay for 12 hours
40                                                          go back to line 8
41         ELSE
42              Check atmospheric temperature and humidity values
43         IF an error is encountered
44                 send SMS error message
45                 STOP

46        ELSE
47                    IF atmospheric temperature is less than atmospheric temperature threshold value AND
humidity values                          are greater than humidity ThV
48                                                          delay for one hour
49                                                          go back to line 42
50                                       ELSE
51              switch on water pump
52              delay for one hour
53               check soil moisture value
54            IF (soil moisture level is less than soil moisture value)
55                 go back to line 51
56            ELSE
57                 Switch off water pump
58                 delay for 12 hours
59                                                          go back to line 2
```

Fig. 3. Smart Irrigation System algorithm

B. System Model

The system consists of soil moisture, temperature, and humidity sensors, a weather monitoring module, and an irrigation actuator. It operates in either automatic or manual mode, with irrigation decisions based on real-time environmental conditions. The system continuously monitors environmental variables and activates irrigation only when predefined conditions are satisfied. The system is defined over continuous time $t \in [0, T]$ with state variables S(t), T(t), H(t) and W(t), which denotes soil moisture level at time t, ambient temperature at time t, relative humidity at time t and weather condition at time t respectively. S_{th} (soil moisture threshold), T_{th} (ambient temperature threshold), (relative humidity threshold), and τ (rain delay time), denoted by RAIN_DELAY, are the system-predefined parameters. The control variables of the system are $M \in \{0, 1\}$, $W(t) \in \{0, 1\}$ and $I(t) \in \{0, 1\}$. M is operation mode, W(t) is weather condition at time t and I(t) is the irrigation decision at time t respectively, which belong to the set of 0 and 1. The binary classifications for M, W(t), and I(t) are represented by equations 1–3, respectively.

$$M = \begin{cases} 1 & \text{Automatic Mode} \\ 0 & \textit{Manual Mode} \end{cases} \quad (1)$$

$$W(t) = \begin{cases} 1 & \text{Rain} \\ 0 & \textit{No Rain} \end{cases} \quad (2)$$

$$I(t) = \begin{cases} 1 & \textit{Irrigation} \text{ ON} \\ 0 & \textit{Irrigation} \text{ OFF} \end{cases} \quad (3)$$

The rain-delay mechanism introduces a time-shifted evaluation in equation 4

$$t_d = t + \tau \quad (4)$$

Where t_d is delay evaluation time. Soil moisture is re-evaluated in equation 5

$$S(t_d) = S(t + \tau) \quad (5)$$

Soil moisture evolves over time based on irrigation and environmental factors and is modelled in equation 6 using a differential equation

$$\frac{dS(t)}{dt} = \alpha I(t) - \beta E(t) + \gamma R(t) \quad (6)$$

α is irrigation efficiency coefficient, β is evaporation rate coefficient, γ is rain contribution coefficient, E(t) is the evaporation function at time t and R(t) is rainfall intensity at time t. Evaporation is influenced by temperature and humidity. Equation 7 modelled the evaporation function at time t.

$$E(t) = k \cdot T(t) \cdot (1 - H(t)) \quad (7)$$

k is the environmental constant and this model captures the fact that higher temperature increases evaporation and higher humidity reduces evaporation.

The rainfall intensity contribution is defined by equation 8, r(t) is rainfall intensity function.

$$R(t) = \begin{cases} r(t), & \text{if } W(t) = 1 \\ 0, & \text{if } W(t) = 0 \end{cases} \quad (8)$$

For this model to work the soil moisture level must remain within safe bounds of $S_{min} \leq S(t) \leq S_{max}$

C. Irrigation Events

During the operation of the system, irrigation treatment is applied to the crop after meeting certain conditions using a rule-based concept approach. The two events involved are shown in Algorithms 1 and 2.

Algorithm 1: Pseudocode for Irrigation event #1

```
Initialise system
Set Threshold_Soil_Moisture
Set Threshold_Temperature
Set Threshold_Humidity
WHILE (Operation_Mode == "Automatic" AND
        Soil_Moisture < Threshold_Soil_Moisture AND
        Weather == "NO_RAIN" AND
        Temperature > Threshold_Temperature AND
        Humidity < Threshold_Humidity)
    Turn ON Irrigate system
  Display "Irrigation in progress..."
  Update Soil_Moisture
  Update Weather
  Update Temperature
  Update Humidity
END WHILE
Turn OFF irrigation system
Display "Irrigation stopped"
```

Algorithm 2: Pseudocode for Irrigation event #2

```
Initialise system
Set Threshold_Soil_Moisture
Set RAIN_DELAY
WHILE (Operation_Mode == "Automatic" AND
     Soil_Moisture < Threshold_Soil_Moisture AND
     Weather == "Rain")
  Wait for Time_Delay = RAIN_DELAY
  Update Soil_Moisture
  IF (Soil_Moisture < Threshold_Soil_Moisture) THEN
     Turn ON Irrigation
  END IF
END WHILE
```

IV. System Implementation

GPS sensors, Arduino UNO and NODEMCU microcontrollers, a real-time clock chip, a mobile SIMCOMM chip, and a pump relay were used in the system's implementation. The design was automated using the Proteus software version 8.8. A scriptable program for electronic design automation, EAGLE (Easily Applicable Graphical Layout Editor) version 9.4.2 is used for the simulation and design of

circuit diagrams. Several components were utilised to implement the concept using the Arduino IDE 1.9.0-beta version. Additionally, the necessary waveforms for the circuit were checked using a HANTEK 6022BL oscilloscope. The digital signals (UART) were compared to the expected signals from each module and the system as a whole using the logic analyser. The sensor unit detected and transmitted the volumetric water content (VWC) of the soil and the environmental conditions, and the system control unit was displayed on the display unit. The data from the sensor and location detection units were processed by the control unit in making decisions and powered by the power supply unit, while the coordinate readings were transmitted to the control unit through the location detection unit. Irrigation activities were done by the irrigation unit whenever an irrigation event was triggered. The processed data was transmitted to the smart irrigation management system, which logged the events on the farmland through the control unit while the user interface was used to communicate with the smart irrigation management system.

A. System Irrigation Prediction Analysis

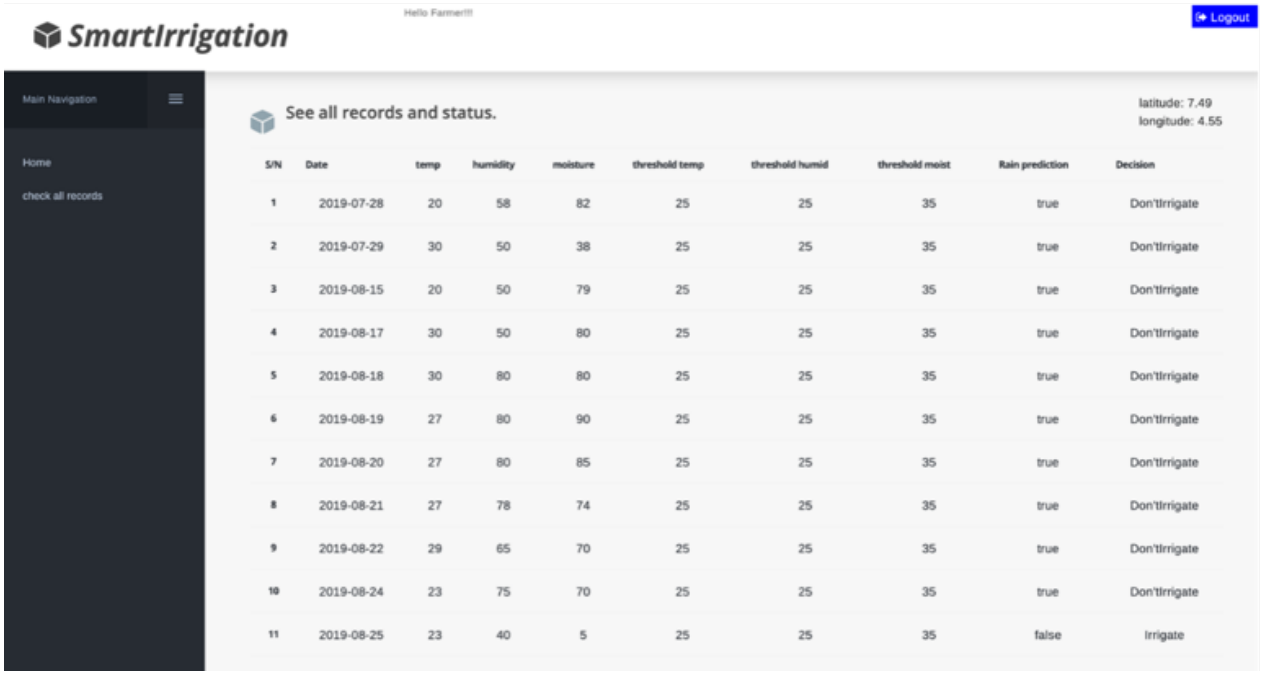
Ile-Ife, Nigeria, where the implementation was tested is predominantly deep clayey soil [16], hard when dry and lumpy and sticky when wet. It also has limited air space and a subpar drainage system. Although it is challenging to cultivate clay soil, plants are known to thrive effectively under appropriate drainage management [17]. Previous research conducted in this area by Emenike and others [16] concluded that clay soil with a volumetric water content (VWC) of 40% to 49% is ideal for supporting the root system of any plant [18]. However, in Ile-Ife, Nigeria, the system used 35% as the threshold value for soil moisture. Additionally, based on observable growth, optimal values for popular crop species, 25°C for temperature and 25% for humidity, were selected [19].

B. Data Gathering and Storage

Through POST and GET operations, the system and the web server exchange data in JSON format, which is then structured into a tabular format and stored in a MySQL database. The threshold values were saved and contrasted with real values using the learning procedure that is detailed in the next section.

C. System Learning Process for Irrigation Prediction

There are three or four steps involved in the system learning process for irrigation: the first compares soil moisture value with a threshold for soil moisture; the second compares the outcome of the first step with a weather forecast; and the third compares air temperature and humidity data values with a threshold for air temperature and humidity, respectively. The system advances to the fourth stage, which compares the new soil moisture data to a threshold value for soil moisture to calculate the ideal water level in the soil if irrigation takes place after step three. Based on data received from all expected occurrences in the system, Fig. 4 shows the irrigation prediction choice.



S/N	Date	temp	humidity	moisture	threshold temp	threshold humid	threshold moist	Rain prediction	Decision
1	2019-07-28	20	58	82	25	25	35	true	Don'tIrrigate
2	2019-07-29	30	50	38	25	25	35	true	Don'tIrrigate
3	2019-08-15	20	50	79	25	25	35	true	Don'tIrrigate
4	2019-08-17	30	50	80	25	25	35	true	Don'tIrrigate
5	2019-08-18	30	80	80	25	25	35	true	Don'tIrrigate
6	2019-08-19	27	80	90	25	25	35	true	Don'tIrrigate
7	2019-08-20	27	80	85	25	25	35	true	Don'tIrrigate
8	2019-08-21	27	78	74	25	25	35	true	Don'tIrrigate
9	2019-08-22	29	65	70	25	25	35	true	Don'tIrrigate
10	2019-08-24	23	75	70	25	25	35	true	Don'tIrrigate
11	2019-08-25	23	40	5	25	25	35	false	Irrigate

Fig. 4. Result of irrigation prediction decision

V. Results And Discussions

The developed system was tested using alpha testing to ascertain the correctness of the system's readings and the beta concept to validate the system for efficiency. The system performance was evaluated using GPS location sensitivity for accuracy and precision, system irrigation prediction analysis for system exactness and system response time. The detailed results are presented as follows:

A. Irrigation Decision Model

Based on the irrigation event algorithm, an irrigation prediction analysis was performed on the system, the result is in Fig. 4. The irrigation system is activated only when the logical control function conditions are satisfied in equation 9.
Equation 9 ensures that irrigation is activated only when the soil moisture is below the required level, there is no rainfall, the temperature is sufficiently high and humidity is low.

$$I(t) = \begin{cases} 1, & \text{if } M = 1 \wedge S(t) < S_{th} \wedge W(t) = 0 \wedge T(t) > T_{th} \wedge H(t) < H_{th} \\ 0, & otherwise \end{cases} \quad (9)$$

Equation 10 incorporates both current and delayed soil moisture conditions.

$$I(t) = \begin{cases} 1, & \text{if } M = 1 \wedge W(t) = 1 \wedge S(t) < S_{th} \wedge S(t+\tau) < S_{th} \\ 0, & otherwise \end{cases} \quad (10)$$

When (5) is substituted in (10), irrigation decision model transforms the system into a time-delay dynamic system in equation 11.

$$I(t) = \begin{cases} 1, & \text{if } M = 1 \wedge W(t) = 1 \wedge S(t) < S_{th} \wedge S(t_d) < S_{th} \\ 0, & otherwise \end{cases} \quad (11)$$

This ensures that irrigation is not triggered immediately during rainfall. The system waits for a delay period τ , and irrigation occurs only if soil moisture remains insufficient after rainfall. Figs. 5–7 show the temperature sequence, humidity, and soil moisture sensor readings obtained from the system irrigation prediction analysis generated over a period through the sensors, respectively. The graphs show the temperature, humidity, and soil moisture sensor values below and above their respective thresholds. Fig. 7 shows that there is only one value below the soil moisture threshold value, corresponding to the only time in Fig. 4 when the system activates the pump for irrigation.

The accuracy and precision of the system were measured using GPS location sensitivity, as well as the system response time, as follows:

1) GPS Location Accuracy

The precision of the meteorological data the system uses depends on how accurately the GPS located the farms. Using the Global Navigation Satellite System (GNSS) evaluation software, the GPS receiver's performance was analyzed for the purposes of the study. The software shows the longitude and latitude coordinates of the farm, along with the available satellites connected to, and Position Dilution of Precision (PDOP) and Horizontal Dilution of Precision (HDOP). Wolf [20] states that the mean DOP (HDOP and VDOP), also known as the PDOP, is a measure of GPS precision. For Open Street Map (OSM) tracking, a number less than 4 denotes a good fix in precision, a value less than 2 denotes the best fix in precision, and a value larger than 6 corresponds to a poor fix. It is only possible to acquire a good DOP in a two-dimensional fix. The GPS demonstrates good precision, and the four satellites are linked, as seen by the PDOP value of 3.7.

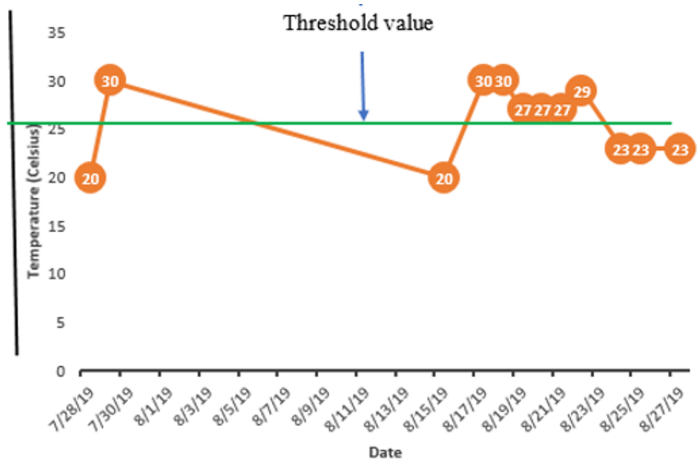


Fig. 5. Temperature values generated over days

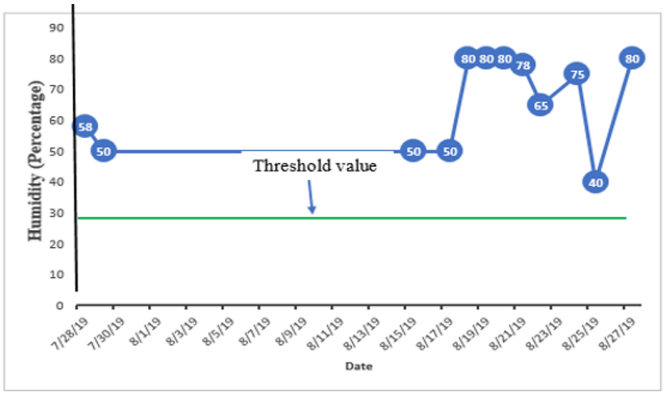


Fig 6. Humidity values generated over days

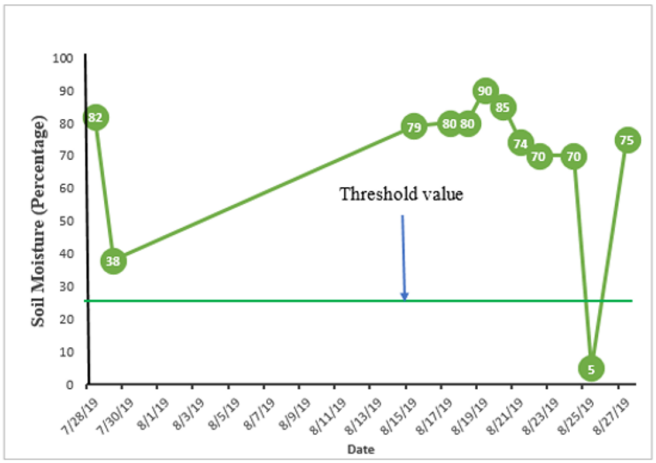


Fig. 7. Soil moisture values generated over days

2) System Response Time

The system's response was determined using the apitester command (GET REQUEST). Fig. 8 depicts the system response time against twenty test samples. The system took 0.830s on average to respond to a GET REQUEST, with minimum and maximum timings of 0.730s and 1.050s, respectively.

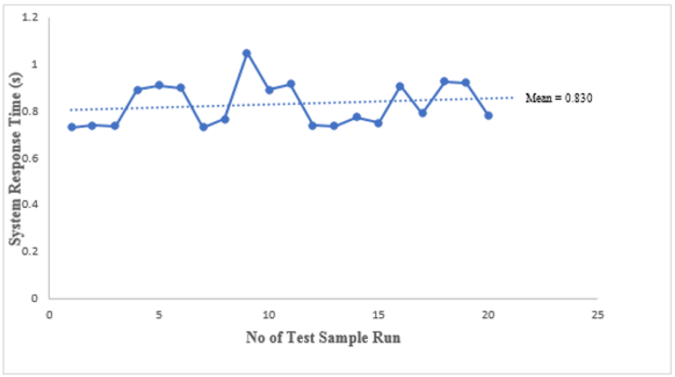


Fig. 8. System response time and test Number of GET REQUEST

B. Irrigation Prediction Accuracy

Fig. 9 shows that the irrigation prediction accuracy for the system is 97%. The study identified and designed Smart Irrigation System modules that are adaptive to climatic conditions, based on irrigation decision models in equations 9 to 11. Irrigation decision models were used to predict the appropriate irrigation status and assess system performance to improve water management and energy efficiency. The system checks soil moisture at time t is < soil moisture threshold, and after delay condition, implements the decision logic in equation 12

$$I(t) = f\big(S(t), S(t+\tau), \mathrm{W(t)}\big) \qquad (12)$$

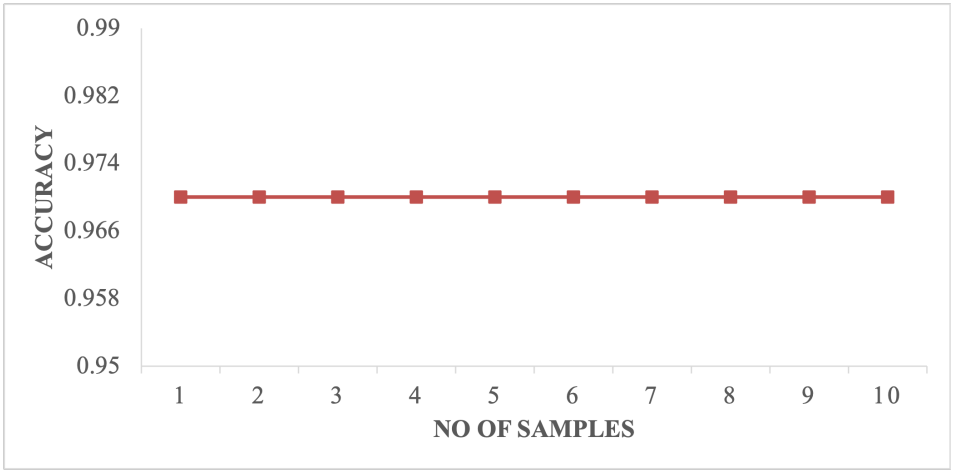


Fig. 9. Irrigation prediction accuracy of the system

VI. CONCLUSION

The system accurately predicts irrigation events by 97%, based on the irrigation decision model and the number of samples fed into irrigation system i.e. data from sensors (GPS, soil moisture, temperature, and humidity) and real-time meteorological information from the farm site. The goal of a climate-adaptive

Smart Irrigation System based on IoT technology is to use water and energy more efficiently and effectively by automating the irrigation structure of the agricultural area in real-time. The system processes the actual data received from farm sensors using a decision model, comparing it to threshold values already set in the SIMS for analysis. The outcome of the comparison to determine whether it will rain or not is always rechecked before making an appropriate decision based on the outcome of the irrigation decision model. The outcome of this research is a weather-sensitive smart irrigation system that minimises labour costs, prevents water loss, and effectively preserves energy. Additionally, it advanced our understanding of the framework for microcontroller-based smart irrigation, which can be extended to a large-scale agricultural environment using a multi-agent systems approach.

REFERENCES

[1] Oni, T. O. (2013). Challenges and Prospects of Agriculture in Nigeria: The Way Forward. *Journal of Economics and Sustainable Development*. 4(16), 37-45. ISSN 2222-1700 (Paper) ISSN 2222-2855 (Online)

[2] Burney, J., Woltering, L., Burke, M., Naylor, R. and Pasternak, D. (2010). Solar-powered drip irrigation enhances food security in the Sudano–Sahel. Proceedings of the National Academy of Sciences, 107(5), 1848-1853. doi:10.1073/pnas.0909678107

[3] Baptista, S., Brottem, L., de Sherbinin, A., Edquist, M., Fischer, A., Levy, M., Schnarr, E., Simon, C., Sundareshwar, P.V. and Trzaska, S. (2013). Background paper for the ARCC West Africa Regional Climate Change Vulnerability Assessment. USAID African and Latin American Resilience to Climate Change (ARCC). http:// community.eldis.org/ARCC

[4] Boansi, D., Tambo, J. A. and Müller, M. (2019). Intra-seasonal risk of agriculturally relevant weather extremes in West African Sudan Savanna. *Theoretical and Applied Climatology*, 135, 355-373. https://doi.org/10.1007/s00704-018-2384-x

[5] Roy, D. K. and Ansari, M. H. (2014). Smart Irrigation Control System. *International Journal of Environmental Research and Development*, 4(4). 317-374. https://www.ripublication.com/ijerd_spl/ijerdv4n4spl_16.pdf

[6] Martin D. L., Dorn, T. W., Melvin, S. R. Corr, A. J. and Kranz W. L. (2011). Evaluating Energy Use for Pumping Irrigation Water. Proceedings of the 23rd Annual Central Plains Irrigation Conference, Burlington, CO., February 22-23, 2011, 104-116. http://hdl.handle.net/10217/47726

[7] Uddin, J., Reza, S. M. T., Newaz, Q., Uddin, J., Islam, T. and Kim, J. (2012). Automated irrigation system using solar power. Proceedings of 7th International Conference on Electrical and Computer Engineering, ICECE 2012; Dhaka; Bangladesh, 20 – 22 December, 2012. DOI: http://dx.doi.org/10.1109/ICECE.2012.6471527

[8] Gutiérrez, J., Villa-Medina, J. F., Nieto-Garibay, A., and Porta-Gándara, M. Á. (2014). Automated Irrigation System Using a Wireless Sensor Network and GPRS Module, *IEEE Transactions on Instrumentation and Measurement*, 63(1), 166-176. DOI: 10.1109/TIM.2013.2276487.

[9] Malge S. and Bhole K. (2015). Novel, Low Cost Remotely Operated Smart Irrigation System. Proceedings of International Conference on Industrial Instrumentation and Control (ICIC), Pune, India, 28-30 May, 2015, 1501-1505. **DOI:** 10.1109/IIC.2015.7150987

[10] Agrawal N. and Singhal S. (2015). Smart Drip Irrigation System using Raspberry pi and Arduino. Proceedings of International Conference on Computing, Communication and Automation (ICCCA), Noida, India, 15-16 May, 2015, 928-932. DOI:10.1109/ccaa.2015.7148526.

[11] Yadav, R., Sharma, M., Azharuddin, M., Monis, M. and Siso, K. (2015). Smart Irrigation System Using GSM and Solar Panel. *International Journal of Scientific Research and Management Studies (IJSRMS)* 3(5), 203-209. ISSN: 2349-3771

[12] Narayut, P., Sasimanee, P., Anupong, C.I., Phon, P. and Khajonpong, A. (2016). A Control System in an Intelligent Farming by using Arduino Technology. Student Project Conference (ICT-ISPC), Fifth ICT International, 53-56, DOI:10.1109/ict-ispc.2016.7519234

[13] Arvindan, A. N. and Keerthika, D. (2016). Experimental investigation of remote control via Android smart phone of Arduino-based automated irrigation system using moisture sensor. Proceedings of 3rd International Conference on Electrical Energy Systems (ICEES), Chennai, India, 17-19 March, 2016, 168-175. DOI: 10.1109/ICEES.2016.7510636.

[14] Kathuria S., Kumar R. and Bhatia T. (2018). E-KRISHI: An IOT Based Smart Irrigation System. *International Journal of Electrical, Electronics and Data Communication,* 6(6), 5331-5333. ISSN(p): 2320-2084, ISSN(e): 2321-2950

[15] Kissoon, D., Deerpaul, H. and Mungur, A. (2017). A Smart Irrigation and Monitoring System. *International Journal of Computer Applications*, 163(8), 39-45. https://www.ijcaonline.org/archives/volume163/number8/kissoon-2017-ijca-913688.pdf

[16] Emenike, H.I., Okeke, H.U., Babalogbon, B.A., Abayomi, T. A. and Ihejirika, G. O. (2015). Land Suitability Assessment for Rice Cultivation in Opa River Basin in Osun State Nigeria Using Multi-Criteria and Geographical Information (GIS) Approach. International Journal of Agriculture and Rural Development. 18(2), 2257-2264.

[17] Salam, A. and Vuran, M. C. (2016). Impacts of Soil Type and Moisture on the Capacity of Multi-Carrier Modulation in Internet of Underground Things. Proceedings of 25th International Conference on Computer Communication and Networks (ICCCN), Big Island, USA, 1-4 August, 2016, 1-9 DOI:10.1109 /ICCCN. 2016. 7568532

[18] Pitts, L. (2016). Monitoring Soil Moisture for Optimal Crop Growth. Online Journal.

[19] Hatfield, J. L. and Prueger, J. H. (2015). Temperature extremes: Effect on plant growth and development. *Weather and Climates Extremes*, 10, 4-10.

[20] Wolf, J. (2004), "Defining GPS and Its Capabilities", Hensher, D., Button, K., Haynes, K. and Stopher, P. (Ed.). Handbook of Transport Geography and Spatial Systems, 5, Emerald Group Publishing Limited,411-431. https://doi.org/10.1108/9781615832538-023