International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Solar-Powered Smart Irrigation System using Machine Learning & IoT

Dasarinki Ramprasad [a], Dhoni Sai Kumar [a], Gujju Siva Reddy [a], Bevara Ravi Teja [a], Dunna Varshitha Sri [a]

[a] GMR Institute of Technology, Rajam, Srikakulam-532127, Andhra Pradesh, India
DOI: https://doi.org/10.55248/gengpi.5.0324.0926

A B S T R A C T

The scarcity of water poses a significant obstacle in the agricultural sector, with conventional irrigation techniques often proving to be inefficient and wasteful. To tackle this issue, a solution is proposed in the form of a smart solar irrigation system using the Internet of Things technology and Random Forest algorithms. This system aims to optimize water usage in agriculture by automating irrigation processes, enhancing the yields of crops, and mitigating the usage of water. The paper outlines a methodology for implementing and designing such a system, which incorporates solar panels, a Raspberry Pi module, sensors, IoT devices, and random forest algorithms. By utilizing data from sensors, the system intelligently controls the water pump by getting notifications to mobile devices, thereby simplifying the irrigation process.

Keywords: Photovoltaic cells, Microprocessor, DHT11 Sensor, Moisture Sensor, Machine learning, Algorithms, Water Management in Agriculture.

1. Introduction

In light of global warming and burgeoning global populations, the agricultural sector faces unprecedented challenges to sustainably produce food. Traditional farming methods are no longer sufficient to meet the growing demand for food while mitigating the impact on the environment. However, advancements in technology offer a torch of hope, providing innovative solutions to enhance agricultural practices. Among these technological innovations, the integration of the Internet of Things and machine learning has emerged as a transformative approach in modern agriculture. One such application is the development of solar-powered smart irrigation systems, which revolutionize the way farmers manage water resources, optimize crop yields, and conserve energy. The convergence of IoT and machine learning in agriculture represents a paradigm shift, empowering farmers with real-time insights and decision-making capabilities. At the heart of this technological synergy lies the concept of precision agriculture, where data-driven strategies are employed to optimize resource utilization and improve crop productivity. By leveraging IoT sensors, which are deployed across agricultural fields, farmers can collect a myriad of data about soil moisture levels, weather conditions, crop health, and environmental parameters. These sensors act as the sensory organs of the farm, continuously monitoring the dynamic ecosystem and relaying valuable information to centralized control systems.

The foundation of a solar-powered smart irrigation system is its ability to harness renewable energy sources to power IoT devices and agricultural machinery. Solar energy, abundant and inexhaustible, offers a sustainable solution to address the energy demands of modern agriculture while reducing reliance on fossil fuels. Photovoltaic (PV) panels, installed strategically within agricultural landscapes, capture sunlight and convert it into electricity, which is then utilized to operate irrigation pumps, sensors, and communication devices. This synergy between solar power and IoT technology not only ensures uninterrupted operation but also minimizes the carbon footprint of agricultural activities. Moreover, machine learning algorithms play a pivotal role in optimizing the performance of smart irrigation systems by analyzing vast amounts of sensor data and identifying patterns and trends. These algorithms employ predictive analytics to forecast crop water requirements, predict weather patterns, detect anomalies, and optimize irrigation schedules. By learning from historical data and adapting to changing environmental conditions, machine learning models empower farmers to make informed decisions that enhance resource efficiency and maximize crop yields. Through iterative optimization and feedback loops, these algorithms continuously refine their predictive capabilities, ultimately leading to more sustainable and resilient agricultural practices.

2. Literature Survey

1. The literature surrounding solar-powered smart irrigation systems incorporating IoT and machine learning underscores the significant strides made in agricultural technology toward sustainability and efficiency. Jain (2023) and Divyapriya et al. (2020) have conducted studies on IoT-enabled drip irrigation systems that utilize weather forecasting, highlighting the pivotal role of data-driven insights in optimizing water management practices. Similarly, Guravaiah and Raju (2020) have explored e-agriculture approaches integrating weather forecasts into irrigation systems, showcasing the potential for improved resource allocation and crop yield prediction [1] [2].

2. Furthermore, research by Vij et al. (2020) and Baba et al. (2020) has focused on IoT and machine learning applications for automating farm irrigation systems, emphasizing the importance of real-time data analytics in enhancing irrigation efficiency and reducing water wastage. Additionally, Abas et al. (2021) and Shah et al. (2021) have examined sustainable energy technologies, particularly solar power, in the context of smart irrigation, highlighting the role of renewable energy sources in powering agricultural systems and reducing carbon footprints [3] [4].

3. Moreover, Veeralakshmi et al. (2022) and Truong et al. (2021) have contributed to the literature by investigating machine learning algorithms for optimizing irrigation schedules based on environmental parameters, further emphasizing the potential for data-driven decision-making in agricultural practices. Finally, Alnaimi et al. (2021) have explored the integration of artificial intelligence and machine learning in agricultural applications, demonstrating the potential for advanced decision support systems to revolutionize smart irrigation technology [5] [6].

3. Proposed Approach:

An IoT-based smart solar irrigation system with a Random Forest algorithm is proposed:

Agriculture can maximize water utilization with a smart solar irrigation system that uses IoT and machine learning algorithms. Automated irrigation, increased crop yields, and decreased water usage are all possible design options for the system. The development and deployment of an intelligent solar irrigation system utilizing machine learning and the Internet of Things is covered in this suggested methodology.

Structure and System Design:

Solar panels, a water pump, an IoT device, sensors, a water storage tank, and machine learning algorithms will all be part of the system. Pumping water to the irrigation area from the water storage tank, electricity will be produced by the solar panels. Temperature, humidity, light, and soil moisture will all be measured by the sensors. The data will be sent to the phone or PC via IoT devices. The data will be analyzed by algorithms of machine learning, which will then be used to determine when to water the crops.

Gathering of Data:

To gather information on soil moisture, temperature, humidity, light, and precipitation, sensors will be buried in the ground. Using IoT devices, the data will be sent to the phone or PC.

Analysis of Data and Machine Learning:

Machine learning methods will be used to examine the sensor data. When deciding when to water the crops, the algorithms for machine learning will be trained to examine the data. The ideal irrigation period will be determined by machine learning algorithms that consider variables including soil moisture, temperature, humidity, raindrops, and light.

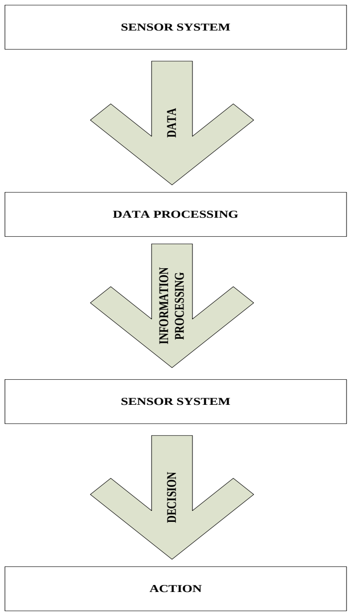

Fig 1: The processing of the Internet of Things

4. Working Methodology:

This project to be designed for a Raspberry Pi-based system that monitors various environmental parameters like temperature, humidity, light intensity, rainfall, and soil moisture. It utilizes sensors such as DHT11 for temperature and humidity, an LDR (Light Dependent Resistor) for light intensity, and

ADS1115 ADC (Analog to Digital Converter) for reading analog sensors like a rain sensor and a soil moisture sensor. Libraries are imported including those for interfacing with sensors (DHT11, LDR, ADS1115), controlling GPIO pins, working with I2C devices (LCD), and for Pushbullet notifications. Initialization of various sensors and components including the LCD, DHT11, ADS1115, LDR, and GPIO pins. Functions are defined for reading sensor values and displaying them on the LCD screen. There are separate functions for reading temperature/humidity, light intensity, rainfall, and soil moisture. These functions also include logic for triggering Pushbullet alerts based on certain conditions being met (e.g., temperature exceeding a threshold, low light intensity, high rainfall, high soil moisture). A while loop continuously runs, calling the sensor reading functions and updating the LCD display with the latest sensor values. Within the loop, there are conditions to check for various alerts and trigger Pushbullet notifications accordingly. The script continues to run indefinitely until manually stopped. This script effectively monitors environmental conditions and provides alerts via Pushbullet when certain thresholds are crossed, allowing for remote monitoring and notification.

5. Results and Conclusions:

Code for working for project:

```
# Include the library files
import I2C_LCD_driver
from time import sleep
import board
import time
import adafruit_dht
from smbus import SMBus
import RPi.GPIO as GPIO
from pushbullet import Pushbullet
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
pb = Pushbullet("o.sD7UwLXkbCkFhtQeAHUWJydQg1YhDGJU")
# Initialize the I2C interface
i2c = busio.I2C(board.SCL, board.SDA)
# Create an ADS1115 object
ads = ADS.ADS1115(i2c)
# Define the analog input channel
channel = AnalogIn(ads, ADS.P0)
channel1 = AnalogIn(ads, ADS.P1)
LDR = 27
bus = SMBus(1)
time.sleep(1)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
# Set the sensor pin as Input pin
GPIO.setup(LDR,GPIO.IN)
# Set the sensor pin as Output pin
GPIO.setup(23, GPIO.OUT, initial=GPIO.LOW)
# Create a object for the LCD
```

```
lcd = I2C_LCD_driver.lcd()
# Create a object for the DHT11 sensor
DHT11 = adafruit_dht.DHT11(board.D4)
# Starting text
lcd.lcd_display_string("System Loading",1,1)
for a in range (0,16):
   lcd.lcd_display_string(".",2,a)
   sleep(0.1)
lcd.lcd_clear()
lcd.lcd_display_string("Wait..",1,0)
def TempHumi():
   try:
      # Get the Temperature and Humidity values
      temperature_c = DHT11.temperature
      temperature_f = temperature_c * (9 / 5) + 32
      humidity = DHT11.humidity
      # Print the values on the LCD display
      lcd.lcd_display_string("T:",1,0)
      lcd.lcd_display_string(str(temperature_c) + ".0C",1,2)
      if temperature_c > 29 :
         dev = pb.get_device('Samsung A52')
         push = dev.push_note("Alert!!", "FARMFIRE")
      lcd.lcd_display_string("H:",2,0)
      lcd.lcd_display_string(str(humidity) + "%",2,2)
      # Print the values to the serial port
      print(
         "Temp: {:.1f} F / {:.1f} C    Humidity: {}% ".format(
            temperature_f, temperature_c, humidity
         )
      )
      except RuntimeError as error:
      # Errors happen fairly often, DHT's are hard to read, just keep going
      print(error.args[0])
      sleep(1)
   except Exception as error:
      DHT11.exit()
      raise error
  sleep(1)
  def light():
```

```
    value = GPIO.input(LDR)
    if value == 0:
        lcd.lcd_display_string("  LIGHT:" + "HIGH",1,8)
      else:
        lcd.lcd_display_string("  LIGHT:" + "LOW ",1,8)
        dev = pb.get_device('Samsung A52')
        push = dev.push_note("Alert!!", "LIGHT LOW")
#Get the analog input values
def rain():
    value2 = channel.voltage*24
    lcd.lcd_display_string("R:",3,0)
    lcd.lcd_display_string(str(value2) + "%          " ,3,3)
    if value2 < 50 :
        lcd.lcd_display_string("HIGH",3,13)
        dev = pb.get_device('Samsung A52')
        push = dev.push_note("Alert!!", "RAINING")
#Get the analog input values
def moisture():
    value1 = channel1.voltage*24
    lcd.lcd_display_string("M:",4,0)
    lcd.lcd_display_string(str(value1) +"%          " ,4,3)
    if value1 < 50 :
        lcd.lcd_display_string("HIGH",4,13)
        dev = pb.get_device('Samsung A52')
        push = dev.push_note("Alert!!", "MOISTURE HIGH")
        GPIO.output(23, GPIO.LOW)
        sleep(1)
    else:
        GPIO.output(23, GPIO.HIGH)
while True:
    TempHumi()
    light()
    rain()
    moisture()
```

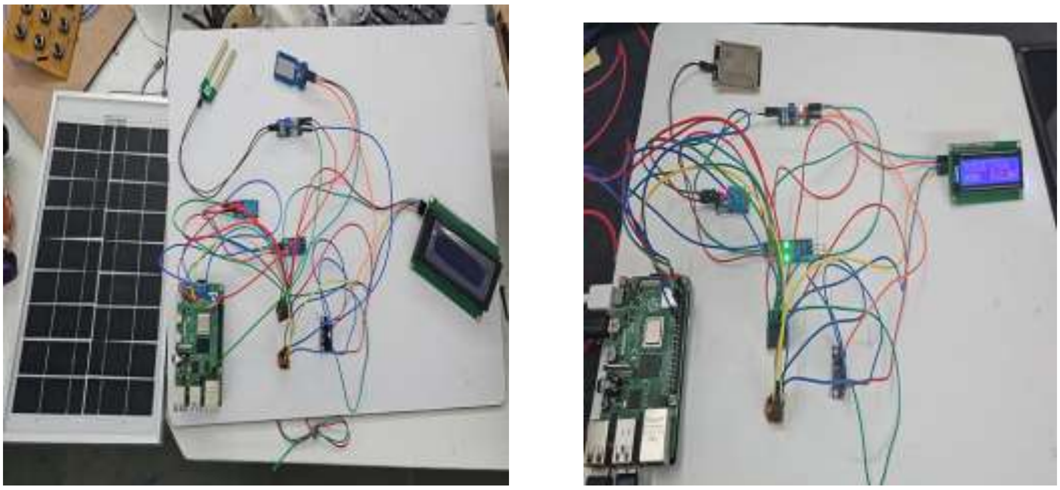

Fig 2: Working Model

Fig 3: Notifications getting from Pushbullet mobile App

REFERENCES

1. Divyapriya, S., et al. "IoT Enabled Drip Irrigation System with Weather Forecasting." 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE, 2020.

2. K. Guravaiah and S. S. Raju, "e-Agriculture: Irrigation System based on Weather Forecasting," 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), RUPNAGAR, India, 2020, pp. 617-622, doi: 10.1109/ICIIS51140.2020.9342739.

3. Vij, Anneketh, et al. "IoT and machine learning approaches for automation of farm irrigation system." Procedia Computer Science 167 (2020): 1250-1257.

4. N. Abas, M. H. Aziz, A. H. Ahmad, M. A. Rahman, and M. R. Islam, 2021 IEEE 6th International Con on Industrial Engineering and Applications (ICIEA), (2021).2.

5. Veeralakshmi, P., Sowmya, S., Kannan, K.N., Anbu, S., Ayyappan, G., (2022), AIP Conference Proceedings, Vol. 2393, no., pp.-.doi:10.1063/5.0074418.

6. H. M. Alnaimi, H. A. Al-Rizzo, and R. K. Al-Rizzo, 2021 IEEE 4th International Con on Artificial Intelligence and Machine Learning Applications (AIMLA), (2021).

7. S. A. Shah, R. J. Jantti, and J. C. Walraven, 2021 IEEE International Con on Sustainable Energy Technologies (ICSET), (2021).

8. N. N. Truong, V. D. Nguyen, V. T. Hoang, and D. D. Nguyen, 2021 IEEE 12th International Con on Intelligent Systems, Modelling and Simulation (ISMS), (2021).
9. Babaa, Saleh Elkelani, et al. "Smart irrigation system using Arduino with solar power." International Journal of Engineering Research & Technology (ijert) 9.05 (2020).
10. Jain, Ravi Kant. "Experimental performance of smart IoT-enabled drip irrigation system using and controlled through web-based applications." Smart Agricultural Technology 4 (2023).