IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412134

Helio Harvest: A Dual-Mode Solar Energy and Rainwater Collection System with ML-Based Water Quality

S Vidhya[1], Pragya[2], Pavithra K[3], Roshni F Gomes[4], V Sandhya[5]

Department of Information Science and Engineering, The Oxford College of Engineering,

Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, India[1-5]

Abstract: The scarcity of water and energy continues to be a major obstacle for sustainable agriculture, necessitating integrated and cost-effective solutions. This project introduces a hybrid framework that incorporates rainwater harvesting, solar energy utilisation, and intelligent automation to optimise resource use in agricultural fields. The cube-structured solar panel system is designed to maximise space efficiency by simulta- neously harvesting rainwater and generating renewable energy. The collected water is filtered and stored in designated tanks, while additional runoff is diverted for future use. A machine learning model predicts water quality in real-time to ensure its safe and effective utilisation by analysing parameters such as pH, turbidity, conductivity, and microbial content. These predictions power an Arduino-based control system that automatically routes water to the field for irrigation or heating for domestic and sterilisation purposes using solar energy. By integrating harvest- ing, prediction and automation, the system increases agricultural sustainability and reduces dependence on conventional resources, and also promises to be highly scalable for both rural and urban applications.

Keywords: Rainwater Harvesting, Solar Energy, Machine Learning, Water Quality Prediction, IoT, Arduino, Sustainable Agriculture

I. INTRODUCTION

Water scarcity is an acknowledged global problem, but it is more critical in India due to rapid urbanisation, climate change, and mismanagement of resources. While Earth is practically a watery planet, only 2.5% of the water is freshwater, and just 1% of the total freshwater is accessible. In India one of the main sources of freshwater is rain. However, due to improper storage and purification mechanisms, a lot of the rainwater resource is wasted. This shortage has subsequently brought about an immediate requirement for better, modern methods related to the capture, purification, and efficient use of rainwater.

The crisis is compounded by the increasing demand for energy. Traditional sources, such as groundwater and electricity from the grid, are usually unreliable or high in environmental cost. Integrating rainwater harvesting with renewable energy, mainly solar power, creates an expedient dual solution capable of satisfying both the water and energy challenges at the same time. Research already supports such integrated systems; proof can be shown that integrated systems provide clean water, renewable energy, and community utility in agricultural fields, campuses, and public open spaces. Hybrid models utilising solar power to purify captured rainwater have been historically supported by literature, showing improved water access and reduced household energy consumption.

II. PROBLEM STATEMENT AND OBJECTIVE

The main objective of the project is for contributing in sustainable water management and the adoption of renewable energy through a minimum-maintenance-designed system for long-term utility. Specific objectives stated for this project include:

1) Design a cube-structured solar panel system that will operate as dual purpose of rainwater harvesting and solar power generation.

2) Automate panel tilting using servo motors and rainfall sensors to optimise rainwater collection efficiency by using the solar panels.

3) Design a multi-stage filtration unit to guarantee that the harvested water is safe for many uses, such as drinking, cooking, irrigation, and domestic use.

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412134

III. LITERATURE REVIEW

[1] K. Alazzam, K. Shatanawi, "Rainwater and fog harvesting from solar panels: Efficiency evaluation under Jordanian climate," *Global Journal of Environmental Science*, 2024.
[2] N. Aryal, T. Regmi, "Investigation of rainwater harvesting technique and use of solar power pump," *ASC: Journal of Applied Science and Technology*, 2023.
[3] L. Bedna´rova´, H. Pavolova´, "Economic efficiency of integrated solar and rainwater systems," *Energies (MDPI)*, 2023.
[4] J. J. John, N. S. Najeeb, "Feasibility of nighttime water harvesting from solar panels," *EPJ Photovoltaics*, vol. 15, 2024.
[5] A. Aktas, S. Sevik, S. Aktas, "Rainwater harvesting in a solar PV power plant: Case study of a 600 kWp installation for irrigation and module cleaning," *Agrivoltaics Clearinghouse*, 2021.
[6] R. K. Sharma, S. K. Singh, "Rainwater harvesting from rooftops: A case study of urban India," *Water Resources Management*, 2022.
[7] P. K. Das, S. Rahman, "Solar-powered water pumping for irrigation: Review and prospects," *Renewable and Sustainable Energy Reviews*, 2021.
[8] S. K. Mishra, A. K. Gupta, "Rainwater harvesting systems for rural com- munities: Design and performance," *Journal of Water Supply: Research and Technology*, 2020.
[9] M. A. Rahman, S. Hossain, "Integrated solar and rainwater harvesting for sustainable agriculture," *International Journal of Energy and Envi- ronmental Engineering*, 2021.
[10] T. K. Das, R. K. Gupta, "Design and implementation of a solar-powered rainwater harvesting system," *Energy for Sustainable Development*, 2020.

IV. SYSTEM ARCHITECTURE

The proposed system is an intelligent, multipurpose setup that integrates rainwater harvesting and the harnessing of *solar energy into a single integrated structure. The core concept is not only to use solar panels for harnessing energy, but also to use them as active rainwater collectors to enable the system to adapt effectively to seasonal changes.*

Physical Design and Dual Functionality: The model includes four solar panels mounted on a servo motor. The proposed system constantly monitors the surroundings through a rainfall sensor.

1) Rainwater Collection Mode: Once rainfall is detected, the automated control system instructs the servo motors to rotate and align the panels into a rectangular container-like setup, similar to an upside-down umbrella. This allows for enhanced efficiency in the capture of rainwater without any wastage due to runoff. The initial runoff is rejected with the support of a first flush system to prevent contaminants from entering the filtration process. The water passes through multiple-stage units, such as 2,000-micron and 20-micrometer meshes, and is stored in separate tanks for further use.

2) Solar Power Generation Mode: During no-rain periods, the panels return to their usual flat position, acting like pho- to voltaic panels to harness sunlight for electricity generation. This forms the renewable energy stored in a 12V battery under the control of a charge controller for powering appliances, lighting (12V LED Strip), or supporting EV charging.

System Automation and Control: The control backbone of the system is based on embedded systems, such as the Arduino Uno and Node MCU, integrated with numerous sensors. Automation is carried out using a tilting mechanism, which consists of servo motors, along with the management of water resources that involves pumps and heating elements.

Machine Learning for Water Quality: The system uses Machine Learning to classify real-time sensor readings automatically into meaningful statuses of water assessment. The attribute vector fed to the learning algorithm includes pH, turbidity, TDS, temperature–humidity, and rainfall sensor readings.

Deployed Model: Lasso Regression
Operational Mechanism: The deployed model is Lasso Regression, adapted for binary classification. When the NodeMCU collects sensor readings, it packages them as a feature vector and sends them to the Flask server. The server loads the pre-trained model file Lasso.sav, performs inference, and generates a numerical prediction. The absolute value of the predicted score is taken and rounded: 0 is Normal (Safe to Use), any non-zero value corresponds to Abnormal (Needs Purification).

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412134

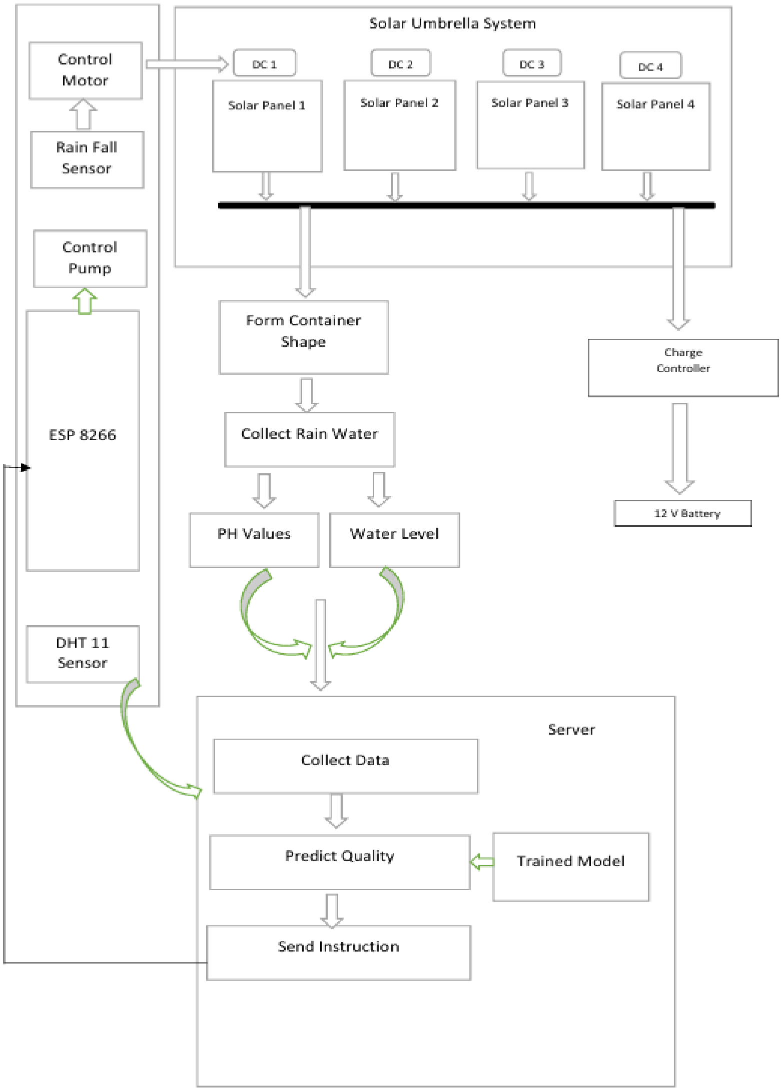


Figure 1 System Architecture Diagram

Integration with System Control: After the Lasso model predicts water status, the Flask backend integrates this output with the real-time mechanical moisture sensor reading. If the model predicts Abnormal and moisture level is high, the system automatically diverts water into the purification unit. When the model predicts Normal, it triggers the valve to direct water straight to the storage tank.

V. MODULE ARCHITECTURE

Central to the hybrid framework is intelligent automation that uses a machine-learning model designed to forecast water
that uses a machine-learning model designed to forecast water quality in real-time, enabling safe and efficient use of har
-vested water.

5.1 Data Acquisition and Preprocessing

The system depends on continuously gathering real-time data from various sensors:

- **pH sensor:** Measures the acid–base balance present in the water
- **Turbidity sensor:** Measures clarity/solids content present in the water
- **Rainfall sensor:** Detects rain intensity and event duration
- **DHT11 Sensor:** Reads temperature and humidity of surroundings
- **TDS sensor:** Measures organic and inorganic content

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412134

Pre-processing steps include:
1) Handling missing/erroneous values by replacing them with average, median and mean values
2) Normalisation of features in a fixed range to ensure stable model performance and reduce redundancy

5.2 Classification Model and Decision Logic

The system uses a pre-trained classification model (such as K-Nearest Neighbors, Random Forest, Decision Trees, XG Boost, or Lasso Regression) that has been trained on previous observations applied to water-quality prediction The training dataset contains features and a 'result' column categorised as 'Normal' (0), 'Moderate' (1), or 'Worst' (2).

If the probability of water being safe exceeds a set threshold, the water is classified as Safe to use and directed into the main storage. Otherwise, the water is classified as Needs Purification, and the flow is redirected through the purification unit before storage. The system supports continuous learning for accuracy and adaptability to local environmental fluctuations. All predictions, with real lab-tested outcomes, are logged. Over time, this data allows retraining and improving the model.

5.3 Architecture Overview

The system includes three integrated subsystems: the IoT sensing and actuation layer, the Flask-based prediction and control engine, and the Android mobile interface for supervision and manual overrides.

Data flows through the system as follows: IoT sensors measure water quality and environmental parameters. The Arduino aggregates raw readings and forwards them to the Node MCU. The Node MCU converts them to a JSON payload and sends them to the Flask server via WiFi. The server receives real-time sensor data and uses the trained Lasso Regression model to predict water status.

The system implements a seven-stage workflow: (1) IoT Sensor Layer captures raw readings from pH, turbidity, TDS, DHT11, and rainfall sensors; (2) Arduino Preprocessing sam- ples ADC, applies calibration, and builds sensor string; (3) NodeMCU Aggregation captures moisture, creates JSON pay- load, and sends HTTP POST; (4) Flask Backend Server receives data and runs Lasso model for prediction; (5) De- cision Logic Engine maps prediction to status, combines with moisture, and selects command; (6) System Control where NodeMCU toggles relays/servos and forwards commands to Arduino; (7) Android App Feedback displays live sensor values, water status, and allows manual commands.

Hardware and Sensing Layer

The hardware module controls sensing and actuation using Arduino Uno and NodeMCU. Sensors include DHT11, pH, turbidity, rainfall, TDS, and mechanical moisture interfaced with the Arduino. For actuation, the system uses MG995 servo motors for mechanical movement and a relay-controlled water pump. Power supply comes from 5V solar panels with a charge controller and a 12V battery.

Network Client Layer

The Node MCU handles Wi-Fi communication: It receives the sensor string from Arduino, appends its own moisture moisture reading, structures everything into a JSON packet, and sends it to the Flask server over HTTP POST.

Backend Server and Prediction Engine

The backend is a Flask server that processes data and generates decisions. The /HelioHarvest route receives JSON payloads from the Node MCU, logs raw readings, extracts numerical values, and feeds them to a trained Lasso Regression model to classify water status as Normal/Abnormal.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412134

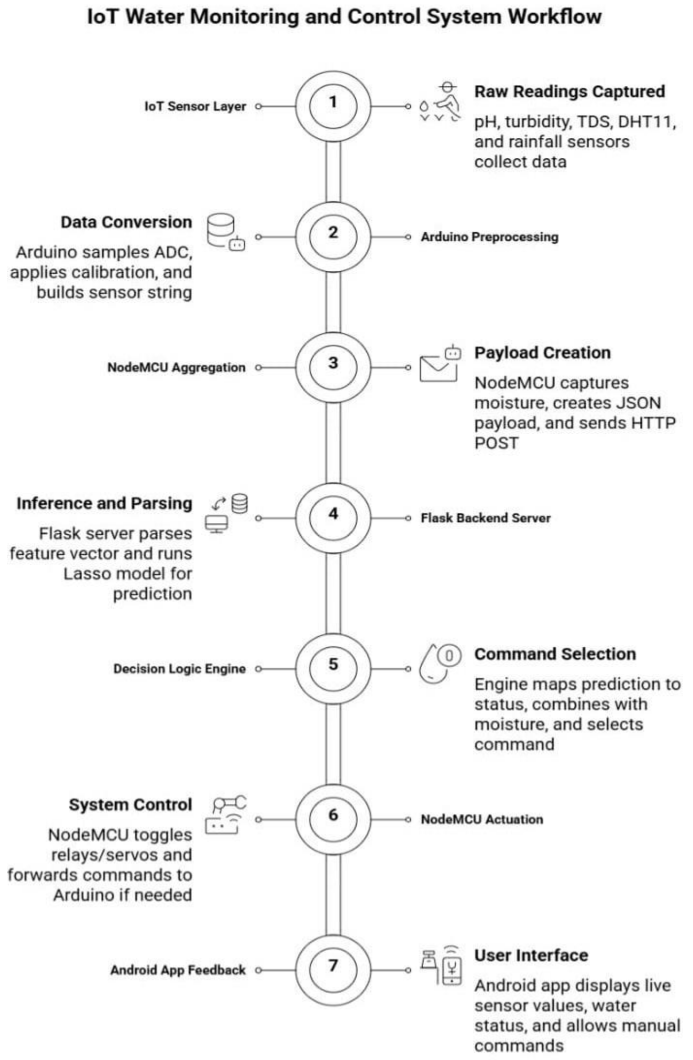


Figure 2 IoT Water Monitoring and Control System Workflow showing the complete data flow from sensors to user interface

VI. IMPLEMENATION ENVIRONMENT

6.1 Hardware Implementation

The hardware setup combines modules of solar energy harvesting, structural control, and environmental sensing. The solar array is composed of 5V photovoltaic panels that feed, through a charge controller, into a 12V battery. MG995 servo motors are for mechanical adjustments. Environmental monitoring includes DHT11 for temperature and humidity, rain sensor for detecting precipitation, and pH and turbidity sensors for checking water quality monitoring.

The microcontroller firmware coordinates sensing, actuation, and data exchange across all system modules. The Arduino/Node MCU platform samples each analog sensor using its internal ADC channels, collecting multiple readings per cycle to suppress random noise. Calibration curves for pH, turbidity, rainfall, and temperature-humidity inputs are applied to convert raw ADC values into engineering units.

Outdoor installation requires the module to be installed in a weather-resistant enclosure with ventilation holes.
The solar panel must be complemented with a charge controller that regulates battery charging to prevent overdischarge.
The wiring and connectors must be rated for moisture exposure.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412134

6.2 Android Application Layer

The Android application represents the main Human–Machine Interface layer, developed in Java using Android Studio. It performs critical functions in terms of remote monitoring, manual control, and configuration of modes.

Application Architecture

The application incorporates:

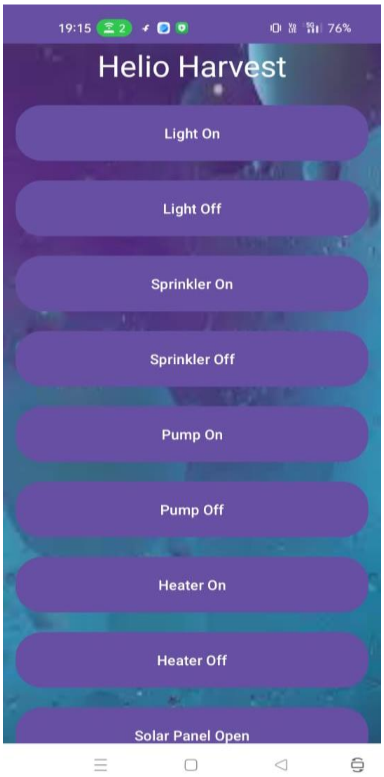
- **Interface Components:** Text Views for operational status and sensor data, Action Buttons for control commands, Mode Button for Manual/Automatic-AI toggle
- **Asynchronous Communication:** Three AsyncTask classes handle networking operations without blocking the UI thread

Real Time Monitoring Mechanism

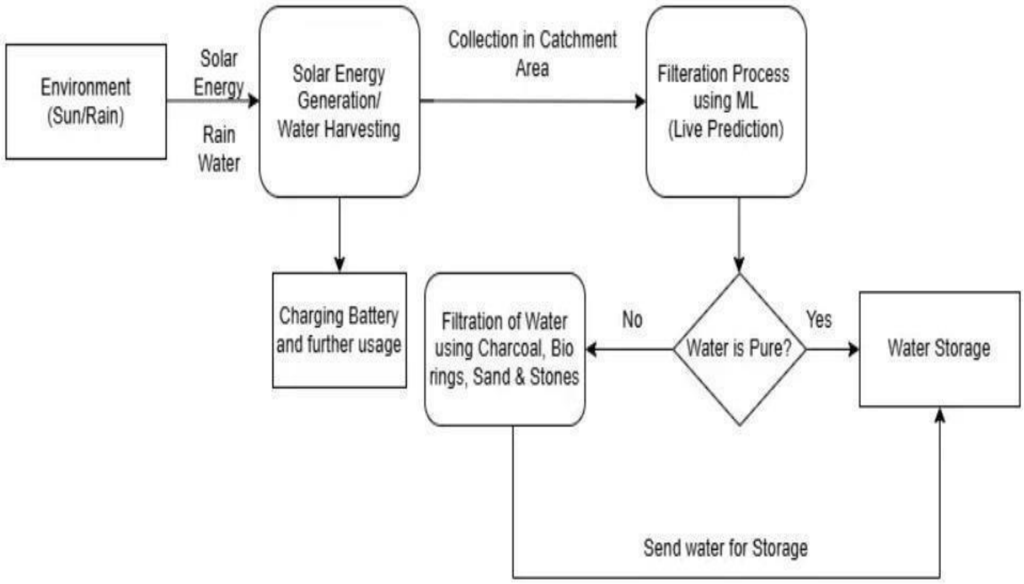
Continuous feedback is managed by Perform Background- Task, which periodically fetches data from the /checkData endpoint every 5 seconds. The server returns latest sensor data and ML prediction status. Measured values of Humidity, Temperature, Rainfall, pH, and Turbidity are extracted and displayed, along with water Status ('Normal' or 'Abnormal').

Manual Control and Mode Configuration

Users can trigger specific hardware operations via labeled buttons. Each button instantiates a Send Instruction Async Task, passing a specific task character for motor control, water management, or lighting. The btn Mode allows toggling between Manual and Automatic-AI modes, sending the appropriate string to the server's /set_mode route.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412134



A Data Flow Diagram shows how information moves inside a system—who sends the data, where it goes, how it is processed, and what actions happen after that. In HelioHarvest, the flow starts with the sensors placed on the system. They collect weather data like rainfall, temperature, and humidity, along with water quality values such as pH, turbidity, and TDS. The Arduino Uno reads this raw data, uses its ADC and calibration formulas, and turns the signals into real, usable values. After that, the calibrated data is sent to the NodeMCU (ESP8266), where it is packed into a small message and sent to the Flask server through Wi-Fi. The Flask server runs the trained Lasso ML model to check the water condition and then creates instruction messages based on the results. These instructions travel back to the NodeMCU and then to the Arduino. Once received, the Arduino runs the necessary hardware actions—like switching the pump, turning on the LEDs, or heating the water through relays.This continuous flow is what makes HelioHarvest run on its own, respond quickly to changes, and stay easy to manage.

VII. MODULES

7.1 NodeMCU/ESP8266: This is a Wi-Fi enabled microcontroller used for communication. The system collects data from sensors like rain sensor, water quality sensors. This data is sent wirelessly to Flask server using ESP8266. The ESP8266 acts like bridge between the hardware setup and cloud. It also receives commands from server and passes them to the Arduino.

7.2 Flask Server: The Flask server works as the backend of the system. It receives sensor data from NodeMCU. The data is processed, checked on server. Based on the values, the server decides what action to take. Instructions are sent back to device so the system runs properly.

7.3 Machine Learning Model: A trained machine learning model is stored on the Flask server. It checks the quality of collected water using values like pH, turbidity, TDS. The model labels water as normal or abnormal. This helps decide whether water can be used or should be released safely.

7.4 Arduino Uno: The Arduino Uno works as the main controller in hardware setup. It reads sensor values using ADC pins. It follows commands received from the server. It controls relays to switch the water pump and other connected parts.

7.5 Rainfall Sensor: This sensor is used to continuously check for rain. It signals Arduino when rainfall is detected, thus activating the servo motors. This, in turn, initiates the tilting or reshaping of the solar panels into a rain collection form and allows the system to automatically switch into water harvesting mode**.**

7.6 Relay Module: This is an electrically operated switch controlled by the Arduino and turn the water pumps, LED strip lights, and water heater ON or OFF through power stored in the 12V battery. Relays ensure higher voltage components' safe handling while enabling automated control.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 12, December 2025
DOI: 10.17148/IJARCCE.2025.1412134

7.7 MG995 Servo Motors: These heavy-duty servo motors are physically attached to the frame of the solar panel. On detecting rainfall throughout the sensor connected to Arduino, servo motors tilt the panel in step-wise mechanical motion towards the formation of a rectangular container-like shape, which allows rainwater to flow toward the inlet for storage through filters.

VIII. PERFORMANCE EVALUATION

8.1 Model Validation Testing

The water-quality classification models were trained on a dataset comprising environmental sensor readings: pH, turbid- ity, TDS, rainfall, temperature, and humidity. Water quality labels (Normal, Moderate, Worst) were encoded as 0, 1, and 2. The dataset was split using a 70:30 ratio for training and testing.

Diverse machine learning techniques were compared based on their cross-validation accuracy. The KNN model (with n_neighbors = 6) provided competitive accuracy. The Decision Tree allowed for faster inference. Ensemble-based models—Random Forest and XGBoost demonstrated better consistency and higher generalisation accuracy. The Lasso model exhibited highly stable behavior under live sensor input fluctuations, produced near-instantaneous predictions, and operated with low computational demand, which made it suitable for deployment as the classifier.

8.2 Hardware Reliability Testing

Multiple live test runs showed the Lasso model was consistent, even with widely fluctuating turbidity and pH values. When the water status was Normal and mechanical moisture level was above threshold, the system directed water to the storage tank. When status became Abnormal, the system correctly sent flow to the purification unit.

8.3 Automated Dual Mode Operation

The Rainfall detection triggered servo motors to rotate the umbrella surface to water-collection configuration. When rainfall subsided, servo control returned the system to solar-panel orientation for photovoltaic energy harvesting. The automated flow-control logic showed expected behavior, correctly routing water based on prediction status.

8.4 Limitations and Considerations

1) Long-term outdoor deployment can wear down servo motors
2) pH and turbidity sensors require periodic recalibration
3) ML model requires more diverse regional datasets for stronger generalization
4) Real-time automation depends on Wi-Fi stability
5) Current setup cannot self-diagnose mechanical faults

8.4 Overall Impact and Deployment Potential

1) Machine learning-driven inference and electromechanical automation work seamlessly
2) System effectively switches between solar-power generation and rainwater harvesting
3) Automated purification routing ensures water safety
4) Low computational load makes system viable in rural environments
5) Framework has promising deployment value in homes, farms, universities and communities

IX. CONCLUSION

The Helio Harvest represents a significant leap in harnessing natural resources by integrating automated rainwater harvesting with solar energy generation under a single intelligent framework. This solution concurrently addresses challenges of water scarcity and energy limitations through integration of electromechanical automation, real-time sensing, machine learning-assisted decision-making, and remote mobile-based supervision.

Success in this project lies in the structural and functional convergence of several technological components:

- Cube-structured solar panel arrangement with dual operating modes
- AI-driven water quality monitoring and control
- Server-side inference-based end-to-end physical actuation control
- Complete Android monitoring and control interface

Unlike traditional systems, the Smart Dual Mode Umbrella transforms surface area into a dynamic multi-purpose asset without resource exclusivity. The implementation of AI-driven water safety prediction instead of fixed threshold-based logic enhances reliability in fluctuating environmental conditions.

This hybrid system has potential to ameliorate several community-level issues:

- Groundwater depletion through extensive rainwater col- lection

DOI: 10.17148/IJARCCE.2025.1412134

- High energy dependency by generating renewable electricity
- Unsafe water reuse through real-time quality classification
- Urban runoff and flooding by capturing rainwater
- Irregular utility access in remote locations

These frameworks contribute to achieving SDG 6 (Clean Water and Sanitation), SDG 7 (Affordable and Clean Energy), SDG 11 (Sustainable Cities and Communities), and SDG 12 (Responsible Consumption and Production).

Future Work

- Mechanical durability improvement for outdoor operation
- Standardized sensor recalibration cycles
- Expanded dataset across different geographical zones
- Reduced Wi-Fi dependency through edge inference
- Advanced purification including UV-C sterilization.

The Smart Dual Mode Umbrella demonstrates that intelligent electromechanical automation with renewable energy and machine learning can transform water and energy management. The successful prototype validates that sustainable technologies need not be expensive, complex, or centralized but could be decentralized, modular, and community-deployable.

ACKNOWLEDGMENT

The authors gratefully acknowledge the faculty and staff of the Department of Information Science and Engineering for their guidance and support throughout this project. We thank our institution for providing laboratory facilities and computational resources.

REFERENCES

[1]. K. Alazzam, K. Shatanawi, “Rainwater and fog harvesting from solar panels: Efficiency evaluation under Jordanian climate,” *Global Journal of Environmental Science*, 2024.
[2]. N. Aryal, T. Regmi, “Investigation of rainwater harvesting technique and use of solar power pump,” *ASC: Journal of Applied Science and Technology*, 2023.
[3]. L. Bedna´rova´, H. Pavolova´, “Economic efficiency of integrated solar and rainwater systems,” *Energies (MDPI)*, 2023.
[4]. J. J. John, N. S. Najeeb, “Feasibility of nighttime water harvesting from solar panels,” *EPJ Photovoltaics*, vol. 15, 2024.
[5]. A. Aktas, S. Sevik, S. Aktas, “Rainwater harvesting in a solar PV power plant: Case study of a 600 kWp installation for irrigation and module cleaning,” *Agrivoltaics Clearinghouse*, 2021.
[6]. R. K. Sharma, S. K. Singh, “Rainwater harvesting from rooftops: A case study of urban India,” *Water Resources Management*, 2022.
[7]. P. K. Das, S. Rahman, “Solar-powered water pumping for irrigation: Review and prospects,” *Renewable and Sustainable Energy Reviews*, 2021.
[8]. S. K. Mishra, A. K. Gupta, “Rainwater harvesting systems for rural com- munities: Design and performance,” *Journal of Water Supply: Research and Technology*, 2020.
[9]. M. A. Rahman, S. Hossain, “Integrated solar and rainwater harvesting for sustainable agriculture,” *International Journal of Energy and Envi- ronmental Engineering*, 2021.
[10]. T. K. Das, R. K. Gupta, “Design and implementation of a solar-powered rainwater harvesting system,” *Energy for Sustainable Development*, 2020.
[11]. A. K. Singh, P. K. Sharmaa, “Machine learning approaches for water quality prediction: A review,” *Environmental Modelling Software*, 2022.
[12]. S. S. Yadav, R. K. Mishra, “IoT-based water quality monitoring and management: Recent trends,” *Sensors*, 2021.
[13]. D. K. Patel, S. K. Singh, “Smart irrigation systems using IoT and machine learning,” *Computers and Electronics in Agriculture*, 2022.
[14]. M. K. Gupta, A. K. Sharma, “Arduino-based automation for water management in agriculture,” *International Journal of Agricultural and Biological Engineering*, 2020.
[15]. S. K. Jain, R. K. Singh, “Fog harvesting: A sustainable water source for arid regions,” *Water Resources*, 2019.