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. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15469

IoT and Machine Learning Based Smart Solar Energy Monitoring and Automatic Dual-Axis Solar Tracking System

G.Ravi[1], B.Lohitha[2]

Assistant Professor, Department of Electronics and Communication Engineering,

Andhra Loyola Institute of Engineering and Technology, Vijayawada, India[1]

Student, Department of Electronics and Communication Engineering,

Andhra Loyola Institute of Engineering and Technology, Vijayawada, India[2]

Abstract: Solar energy is one of the most promising renewable energy sources, yet its efficient utilization remains challenging due to fixed panel orientations, lack of real-time performance monitoring, and absence of intelligent data analysis. This paper presents an IoT and Machine Learning (ML) based Smart Solar Energy Monitoring and Automatic Dual-Axis Solar Tracking System. The proposed system integrates an Arduino UNO microcontroller and an ESP32 module with voltage, current (ACS712), and Light Dependent Resistor (LDR) sensors to continuously measure solar panel output parameters and automatically adjust panel orientation for maximum sunlight absorption. Sensor data comprising voltage, current, power, and energy generation is transmitted to the ThingSpeak cloud platform via Wi-Fi for real-time remote visualization. A dual-axis tracking mechanism employing four LDR sensors and two servo motors ensures continuous alignment of the solar panel with the sun, enhancing energy capture by approximately 30–40% compared to fixed installations. A Linear Regression-based machine learning model developed in Python predicts daily and monthly energy production and estimates potential income, enabling proactive energy management. A Streamlit-based web dashboard provides an interactive interface for real-time and historical data analysis. Experimental results confirm accurate parameter detection, effective solar tracking, reliable cloud data transmission, and practical income prediction. The system is cost-effective, scalable, and suitable for residential rooftop installations, smart solar farms, and educational applications.

Keywords: IoT, Arduino UNO, ESP32, solar energy monitoring, dual-axis solar tracking, LDR sensor, ACS712, ThingSpeak, machine learning, linear regression, Streamlit dashboard.

I. INTRODUCTION

Solar energy is one of the most abundant and renewable sources of energy available on Earth. It is derived from the radiation emitted by the sun and can be converted into electrical energy using photovoltaic (PV) technology. With the increasing demand for energy and the depletion of conventional fossil fuels such as coal, petroleum, and natural gas, solar energy has emerged as a sustainable alternative for power generation. The sun produces an enormous amount of energy every second, and even the small fraction reaching the Earth's surface is sufficient to meet global energy needs if efficiently harnessed.

Despite its potential, traditional solar power systems face several limitations that reduce their efficiency and effectiveness. The most prominent issue is the fixed orientation of solar panels, which prevents them from capturing maximum sunlight as the sun moves across the sky during the day. Additionally, most conventional systems lack real-time monitoring of panel performance, making it difficult to detect faults, measure system efficiency, or analyze power generation trends. The absence of intelligent data analysis further limits accurate prediction of energy production and potential economic benefits.

In countries like India, which receive abundant sunlight throughout the year, solar energy has great potential. Large-scale solar farms, rooftop solar installations, and off-grid systems are being widely implemented. Combining IoT-based monitoring with automatic solar tracking and machine learning analytics can significantly improve system efficiency, reliability, and maintenance.

The proposed system addresses these limitations by integrating an Arduino UNO microcontroller and an ESP32 module with dedicated sensors for voltage, current, and light intensity measurement. A dual-axis solar tracking mechanism using LDR sensors and servo motors continuously aligns the panel toward the sun. Sensor data is uploaded to the ThingSpeak IoT cloud platform for real-time remote monitoring. A Python-based Linear Regression model predicts future energy generation and potential income, while a Streamlit dashboard provides an interactive visualization interface. This paper describes the system design, hardware and software components, experimental results, and future scope.

DOI: 10.17148/IJARCCE.2026.15469

A. Motivation of the Research

Increasing industrial activities, urbanization, and population growth have intensified the demand for clean energy. While solar PV systems offer a viable solution, their efficiency is heavily dependent on panel orientation relative to the sun and the ability to monitor and respond to changing environmental conditions in real time. Traditional fixed systems and manual monitoring approaches are inherently reactive. The motivation of this work is to develop a cost-effective, intelligent, and automated solar energy system that continuously tracks the sun, monitors electrical parameters in real time, uploads data to the cloud, and applies machine learning to forecast energy generation and financial returns.

B. Objectives of the Work

The primary objectives of this project are: (i) to accurately measure key electrical parameters voltage, current, power, and energy generated by the solar panel; (ii) to implement a dual-axis automatic solar tracking mechanism using LDR sensors and servo motors for maximum energy harvesting; (iii) to develop an IoT-based cloud monitoring system using the ThingSpeak platform for real-time remote data visualization; (iv) to apply machine learning (Linear Regression) for predicting solar energy output and potential income; and (v) to provide a user-friendly real-time dashboard using Streamlit for performance monitoring and analysis.

II. LITERATURE SURVEY

Extensive research has been conducted on IoT-based solar monitoring, solar tracking systems, and the application of machine learning in renewable energy management. Smith et al. [1] proposed a real-time IoT monitoring system for solar panels that enabled remote performance tracking; however, the system lacked machine learning-based predictive capabilities. Kumar et al. [2] demonstrated significant efficiency improvements through a solar tracking mechanism but did not integrate cloud-based monitoring. Lee et al. [3] applied machine learning techniques for energy forecasting; their approach was limited by a small dataset, reducing generalization capability. Sharma et al. [4] combined IoT with ML for smart analytics in solar installations, though the high implementation cost restricted practical adoption.

Solar tracking systems are broadly classified into single-axis and dual-axis configurations. Single-axis trackers rotate the panel along one axis (typically east-to-west) and are widely used in medium-scale installations due to their mechanical simplicity. Dual-axis trackers allow rotation along both horizontal and vertical axes, enabling the panel to maintain optimal orientation throughout the day and across seasons, thereby maximizing solar energy capture [5], [6]. Studies show that dual-axis tracking can significantly improve energy output compared to fixed systems. Modern tracking systems increasingly incorporate IoT, GPS, and artificial intelligence for enhanced accuracy and efficiency [7], [8].

Fixed solar panel systems, while simple, cost-effective, and requiring minimal maintenance, suffer from reduced energy efficiency since they cannot adjust orientation according to the sun's changing position. Research confirms that fixed panels often generate substantially less energy than their tracking counterparts [9]. The literature collectively highlights the need for an integrated system that combines continuous real-time monitoring, automatic dual-axis tracking, cloud connectivity, and intelligent predictive analytics — gaps that the proposed system addresses.

III. METHODOLOGY

A. System Overview

The proposed system architecture consists of five functional layers: (i) energy generation layer — solar panel converting sunlight into DC electrical energy; (ii) sensing layer — voltage sensor, ACS712 current sensor, and four LDR sensors; (iii) processing layer — Arduino UNO microcontroller for data acquisition, power calculation, and tracking control, with ESP32 for IoT communication; (iv) communication layer — Wi-Fi-based HTTP data transmission to the ThingSpeak cloud platform; and (v) analytics layer — Python Linear Regression model with Streamlit dashboard for prediction and visualization. Two servo motors constitute the actuation subsystem for dual-axis panel orientation.

B. System Workflow

The operational flow begins with the solar panel converting sunlight into electrical energy. Voltage and current sensors measure the panel output and transmit signals to the Arduino UNO, which calculates instantaneous power using the relation $P = V \times I$ (Eq. 1) and accumulates energy units over time. Simultaneously, four LDR sensors continuously monitor sunlight intensity from different directions. The microcontroller compares LDR readings to determine the direction of maximum sunlight and sends PWM control signals to two servo motors to adjust the panel's azimuth and elevation angles. Processed data is forwarded to the ESP32 module, which uploads it to the ThingSpeak cloud via HTTP GET requests every 3 seconds. The Python ML model retrieves stored data from ThingSpeak, applies Linear Regression, and predicts future energy output and income, displayed on the Streamlit dashboard.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15469

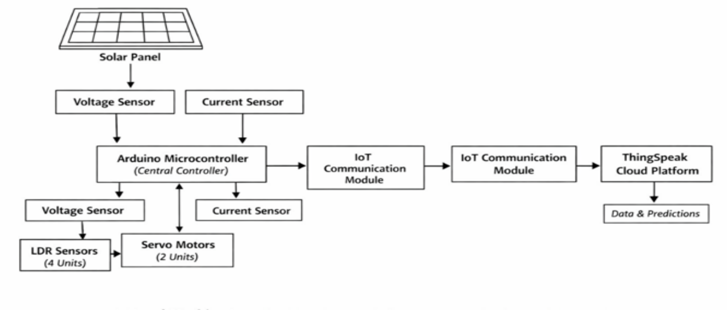

Fig. 1. Functional Block Diagram of the Proposed System.

C. Block Diagram Description

As illustrated in Fig. 1, the solar panel connects to the voltage and ACS712 current sensors. Sensor outputs feed into the Arduino UNO (central controller), which also receives LDR inputs and controls servo motors for tracking. The Arduino communicates serially with the ESP32, which handles Wi-Fi connectivity and cloud data upload to the ThingSpeak IoT platform. A 16×2 I2C LCD displays real-time parameter values on-site. The ThingSpeak cloud stores data, which the Python ML model accesses for analysis and income prediction, displayed on the Streamlit dashboard.

D. Existing vs. Proposed System

Traditional solar installations employ fixed panels with no real-time monitoring, requiring periodic manual inspection and lacking any predictive capability. SCADA-based systems, while capable of data logging, are expensive and reactive rather than proactive. The proposed IoT-based system provides continuous automated monitoring, automatic dual-axis tracking, real-time cloud data logging, remote access from any location, and ML-driven income prediction — substantially reducing human intervention, improving energy yield by 30–40%, and enabling informed energy management decisions.

IV. HARDWARE COMPONENTS

A. Solar Panel

A photovoltaic solar panel composed of multiple silicon PV cells serves as the primary energy source. When sunlight strikes the PV cells, photons excite electrons via the photovoltaic effect, generating a direct current (DC). The magnitude of voltage and current output depends on sunlight intensity and panel efficiency. In this project, the solar panel is mounted on the dual-axis tracking structure and its output is continuously monitored by the sensing subsystem.

B. Arduino UNO Microcontroller

The Arduino UNO (ATmega328P, 16 MHz, 32 KB flash, 14 digital I/O pins, 6 analog input pins, 5 V operating voltage) acts as the central processing unit of the hardware subsystem. It reads analog signals from the voltage, current, and LDR sensors; calculates power and energy; controls servo motors via PWM signals; updates the 16×2 I2C LCD display; and manages serial UART communication with the ESP32 module.

C. ESP32 Microcontroller (IoT Communication Unit)

The ESP32 is a dual-core, 32-bit microcontroller with built-in Wi-Fi (802.11 b/g/n) and Bluetooth capabilities. It receives processed sensor data from the Arduino UNO via UART serial communication and transmits it to the ThingSpeak cloud platform using HTTP GET requests. Its built-in Wi-Fi module eliminates the need for an external communication module, making it highly suitable for IoT applications requiring real-time data upload.

D. ACS712 Current Sensor

The ACS712 current sensor operates on the Hall Effect principle, enabling accurate and electrically isolated measurement of DC and AC current. When current flows through the embedded conductor, it generates a proportional magnetic field detected by the Hall element, which outputs a corresponding analog voltage. The microcontroller reads this analog output and converts it to a current value. The ACS712 (5A variant) offers high accuracy, electrical isolation for component safety, and is used to measure the solar panel's output current for power calculation.

E. Voltage Sensor

The voltage sensor employs a resistive voltage divider circuit to scale down the solar panel's output voltage to a range compatible with the microcontroller's analog input pin (0–5 V). The microcontroller reads the scaled analog value and applies the voltage divider ratio (approximately 4.17) to compute the actual panel voltage. This measurement is essential for calculating power and energy generation in real time.

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. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15469

F. LDR Sensors

Four Light Dependent Resistors (LDRs) are strategically positioned around the solar panel to detect sunlight intensity from different directions. LDRs exhibit an inverse relationship between incident light intensity and electrical resistance — higher illumination produces lower resistance, allowing greater current flow. By comparing analog readings from all four LDRs (Top-Left, Top-Right, Bottom-Left, Bottom-Right), the microcontroller determines the direction of maximum sunlight and sends corresponding control signals to the servo motors for panel alignment.

G. Servo Motors

Two servo motors constitute the actuation subsystem for dual-axis solar tracking. Servo motors operate on a closed-loop control principle, adjusting angular position based on PWM control signals from the microcontroller, where pulse width determines the exact shaft angle. The horizontal servo controls azimuth rotation (east-west direction), while the vertical servo adjusts the elevation angle. This coordinated dual-axis movement enables the solar panel to maintain optimal alignment with the sun throughout the day and across seasons, maximizing energy harvest.

H. Power Supply

A regulated power supply based on the LM7805 voltage regulator provides a stable 5 V DC output from a 12 V input, ensuring consistent operation of the Arduino, sensors, and LCD. The ESP32 and servo motors operate at their respective required voltage levels (3.3 V and 5–6 V). Voltage regulation protects sensitive electronic components from fluctuations, overvoltage, and potential damage.

I. Mechanical Mounting Structure

The mechanical mounting structure supports the solar panel and enables controlled dual-axis rotation. It comprises the panel frame, servo motor mounts, rotating joints, and a sturdy support base. The rotating joints allow smooth angular movement along both horizontal and vertical axes. The structure is designed to balance flexibility and stability, ensuring precise and vibration-free movement critical for accurate solar tracking under varying environmental conditions such as wind.

V. SOFTWARE IMPLEMENTATION

A. Arduino Firmware

The Arduino firmware, developed in the Arduino IDE, serves as the central control program for all hardware operations. It executes continuously in a main loop performing the following tasks: (i) reading analog voltage from the voltage sensor (pin A3) and battery sensor (pin A1), applying the divider ratio to compute actual voltage; (ii) reading the ACS712 current sensor output (pin A2) and computing current as $(V\_sensor - 2.5) / 0.185$; (iii) calculating instantaneous power ($P = V \times I$) and accumulating energy in kWh over elapsed time; (iv) reading four LDR sensor values (pins A0–A3 on tracking controller); (v) comparing LDR averages to determine sunlight direction and adjusting servo motor positions via PWM; (vi) displaying voltage, current, power, and energy on the 16×2 I2C LCD; and (vii) transmitting all parameters to the ESP32 via UART serial communication every 3 seconds.

B. ESP32 Firmware

The ESP32 firmware connects to the designated Wi-Fi network (SSID and password configured at compile time) and initializes the ThingSpeak client with the channel ID and Write API key. In the main loop, it listens for incoming serial data from the Arduino, parses the comma-separated values for voltage, current, power, energy, and battery voltage, and uploads them to ThingSpeak Fields 1–5 respectively using the ThingSpeak.setField() and ThingSpeak.writeFields() functions. A 3-second delay ensures compliance with ThingSpeak's rate limiting. The ESP32's built-in Wi-Fi eliminates the need for an external module, simplifying circuit design.

C. ThingSpeak Cloud Integration

ThingSpeak is a MathWorks-hosted IoT analytics platform for collecting, storing, and visualizing sensor data. A dedicated channel is configured with five fields: Field 1 (Solar Voltage), Field 2 (Solar Current), Field 3 (Solar Power), Field 4 (Energy Units), and Field 5 (Battery Voltage). The ESP32 transmits data via HTTP GET requests containing the Write API key and field values. ThingSpeak renders real-time graphical dashboards for each field, enabling remote monitoring from any internet-connected device. Historical data stored on ThingSpeak is exported in CSV format for machine learning analysis.

D. Python Machine Learning Model

The ML model is developed in Python using the Pandas, NumPy, Scikit-learn, and Matplotlib libraries. Solar energy data exported from ThingSpeak is imported, cleaned, and preprocessed to remove missing or inconsistent values. Features selected as input variables include voltage (V), current (A), power (W), time of measurement, and sunlight intensity (from LDR readings). The target variable is energy generation in kilowatt-hours (kWh). The dataset is split into 80% training and 20% testing subsets. A Linear Regression model is trained on historical data to learn relationships between input

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. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15469

features and energy output, represented by the equation shown in Eq. 2. Model performance is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) metrics. After training, the model predicts future energy generation and estimates potential income using the formula in Eq. 3.

$$\mathbf{P = V \times I} \quad \text{(Eq. 1)}$$
$$\mathbf{Energy = a + b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_nx_n} \quad \text{(Eq. 2)}$$
$$\mathbf{Income = Energy \times Electricity\ Rate} \quad \text{(Eq. 3)}$$

E. Streamlit Dashboard

A Streamlit-based web dashboard provides an interactive real-time interface for system monitoring and analysis. The dashboard retrieves live data from the ThingSpeak cloud and presents dynamic charts and graphs for voltage, current, power generation, and energy units. The left panel displays ML input parameters (voltage, current, power, energy, time, price per unit), while the right panel shows live IoT data from ThingSpeak alongside predicted income values. The dashboard also displays estimated daily and monthly income based on ML predictions, enabling users to assess the economic benefits of their solar installation remotely through any web browser.

VI. RESULTS AND DISCUSSION

A. System Operational Flow

The complete operational flow of the system proceeds as follows: the solar panel generates electrical energy; voltage and current sensors measure the output; the Arduino UNO processes sensor data, calculates power and energy, and controls the tracking servo motors based on LDR comparisons; data is transmitted to the ESP32, which uploads it to ThingSpeak; the Streamlit dashboard visualizes live data; and the Python ML model predicts future energy generation and income. The system operates continuously without human intervention, with the LCD providing on-site feedback even without internet connectivity.

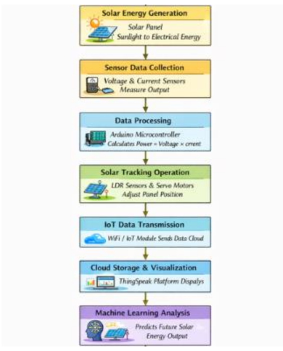

Fig. 2. System Operational Flowchart.

B. Voltage Results

The solar panel output voltage was continuously monitored under varying sunlight conditions. Observations confirm that voltage output is directly proportional to sunlight intensity. Peak voltage values were recorded between 11:00 AM and 2:00 PM, coinciding with maximum solar radiation, while lower values were observed during early morning and late evening. Table II presents sample voltage readings recorded during system testing

TABLE I. SAMPLE VOLTAGE READINGS

Time	Voltage (V)
9:00 AM	12.5 V
11:00 AM	17.8 V
1:00 PM	18.5 V
3:00 PM	16.2 V
5:00 PM	13.4 V

C. Current Results

The ACS712 current sensor measured solar panel output current under varying solar radiation conditions. Current output varies directly with sunlight intensity, with maximum values recorded around midday. These results confirm that current

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁚ Peer-reviewed & Refereed journal ⁚ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15469

generation is highly dependent on solar irradiance and directly determines overall power output. Table III presents sample current readings.

TABLE II. SAMPLE CURRENT READINGS

Time	Current (A)
9:00 AM	0.45 A
11:00 AM	0.82 A
1:00 PM	0.95 A
3:00 PM	0.73 A
5:00 PM	0.40 A

D. Power Generation Results

Instantaneous power was calculated using P = V × I (Eq. 1). Power output varied proportionally with both voltage and current, reaching maximum values during peak sunlight hours. This analysis highlights the direct dependency of power generation on solar radiation and demonstrates the effectiveness of real-time monitoring in evaluating system performance. Table IV presents sample power output data.

TABLE IV. SAMPLE POWER OUTPUT

Time	Voltage (V)	Current (A)	Power (W)
9:00 AM	12.5	0.45	5.6 W
11:00 AM	17.8	0.82	14.6 W
1:00 PM	18.5	0.95	17.6 W
3:00 PM	16.2	0.73	11.8 W
5:00 PM	13.4	0.40	5.3 W

E. Energy Generation and Solar Tracking Performance

Total energy generated was calculated by accumulating power output over time. Under standard test conditions, the solar panel produced an average of approximately 0.12 kWh to 0.18 kWh per day. Energy production varied with weather conditions, sunlight intensity, and daily duration of sunlight exposure. The implementation of the dual-axis solar tracking mechanism significantly improved energy capture compared to a fixed solar panel configuration. By continuously aligning the panel toward the sun using LDR feedback and servo motor actuation, the system increased energy generation by approximately 30–40%.

Fig. 3. Real-Time Solar Data Visualization on ThingSpeak Cloud Platform.

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. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15469

Fig. 4. Assembled Prototype with Tracking Frame and Controller Output.

F. Machine Learning Income Prediction Results

The Linear Regression model was trained on historical solar data collected from the ThingSpeak platform. The model demonstrated reliable predictive performance, as evaluated by MAE, MSE, and R^2 metrics. As an illustrative example, with a daily energy production of 0.15 kWh and an electricity tariff rate of ₹6 per kWh (Eq. 3): Daily Income = 0.15 × 6 = ₹0.90. Estimated monthly income = ₹0.90 × 30 = ₹27.00. While income from a single small-scale panel is modest, the approach scales considerably for larger installations. The ML Prediction Output displayed on the Streamlit dashboard allows users to input custom voltage, current, time, and price parameters to obtain dynamic income forecasts.

Fig. 5. ML Prediction Output on Streamlit Dashboard.

TABLE V. MACHINE LEARNING MODEL PARAMETER SPECIFICATIONS

Parameter	Description
Timestamp	Time at which the data is recorded
Voltage (V)	Output voltage from the solar panel
Current (A)	Current generated by the solar panel
Power (W)	Power calculated from voltage and current
Energy (kWh)	Total energy generated over time
Sunlight Level (LDR)	Light intensity detected by LDR sensors

G. System Advantages

The proposed system provides the following key advantages: (i) real-time continuous multi-parameter monitoring with cloud-based remote access; (ii) automated dual-axis solar tracking improving energy yield by 30–40% over fixed installations; (iii) ML-driven income prediction enabling proactive energy and financial management; (iv) cost-effective and scalable design suitable for residential and commercial deployments; (v) on-site LCD display for local monitoring without internet dependency; and (vi) user-friendly Streamlit dashboard for accessible performance visualization.

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15469

VII. CONCLUSION

This paper has presented an IoT and Machine Learning based Smart Solar Energy Monitoring and Automatic Dual-Axis Solar Tracking System. The system integrates voltage, current (ACS712), and LDR sensors with an Arduino UNO microcontroller and ESP32 Wi-Fi module to deliver continuous automated monitoring of solar panel parameters, real-time cloud data upload to ThingSpeak, and automatic panel orientation control via dual-axis servo tracking. Experimental results confirm accurate parameter detection, effective solar tracking, and reliable cloud data transmission. The implementation of dual-axis tracking increased energy generation by approximately 30–40% compared to fixed installations. The Python Linear Regression model demonstrated practical income prediction capability, and the Streamlit dashboard provided an accessible and interactive monitoring interface.The system is cost-effective, easily replicable, and suitable for residential rooftop solar systems, smart solar farms, research laboratories, and educational institutions. It provides a practical and scalable solution for maximizing solar energy utilization, enabling remote performance monitoring, and supporting informed energy management and financial planning.

VIII. FUTURE WORK

Several enhancements are planned to further improve system performance and scalability. Expansion of the system for large-scale solar farm management through a centralized IoT platform will enable coordinated control of multiple panels. Integration of advanced machine learning models (e.g., Random Forest, LSTM neural networks) will improve prediction accuracy for energy generation, weather impact analysis, and income forecasting. Real-time weather forecasting API integration will allow the system to anticipate energy generation variations based on cloud cover and temperature data. A dedicated mobile application with push notifications will improve user accessibility for real-time monitoring and alerts. Battery storage management integration will optimize energy utilization during periods of low solar generation. Finally, smart grid integration will enable bi-directional energy flow management and net metering support, significantly expanding the system's practical impact.

REFERENCES

[1] Smith et al., "IoT-based real-time solar monitoring system," International Journal of Renewable Energy Systems, vol. 12, no. 3, pp. 45–53, 2020.
[2] Kumar et al., "Solar tracking system for improved photovoltaic efficiency," International Journal of Advanced Engineering Research, vol. 15, no. 2, pp. 112–119, 2021.
[3] Lee et al., "Machine learning for solar energy forecasting," Renewable and Sustainable Energy Reviews, vol. 88, pp. 201–210, 2019.
[4] Sharma et al., "Smart analytics for solar energy using IoT and ML," Smart Innovation, Systems and Technologies, vol. 102, pp. 220–230, 2022.
[5] M. Kanoglu, Y. A. Çengel, and J. M. Cimbala, Fundamentals and Applications of Renewable Energy, 1st ed. New York, NY, USA: McGraw-Hill, 2020, pp. 1–15.
[6] G. Prinsloo and R. Dobson, Solar Tracking: Sun Position and Sun Following. South Africa: SolarBooks, 2015, pp. 3–33.
[7] R. F. Fuentes-Morales et al., "Control algorithms applied to active solar tracking systems: A review," Solar Energy, vol. 212, pp. 203–219, Dec. 2020.
[8] K. Kumar et al., "Solar tracker review," International Transactions on Electrical Energy Systems, vol. 31, no. 12, 2021.
[9] M. I. Al-Amayreh and A. Alahmer, "Improving efficiency using dual-axis solar tracking," Energy Reports, vol. 8, pp. 841–847, 2022.
[10] S. Ozcelik, H. Prakash, and R. Challoo, "Two-axis solar tracker analysis and control for maximum power generation," Procedia Computer Science, vol. 6, pp. 457–462, 2011.
[11] N. G. Hariri et al., "Experimental investigation of azimuth and sensor-based control strategies," Applied Sciences, vol. 12, no. 9, 2022.
[12] M. Dehghan, S. Rashidi, and A. Waqas, "Modeling of soiling losses in solar energy systems," Sustainable Energy Technologies and Assessments, vol. 53, Oct. 2022.
[13] D. Mazumdar et al., "Design of LQR controller for solar tracking system," in Proc. IEEE ICECCT, 2015, pp. 1–5.
[14] M. Ikhwan and C. Imron, "Model predictive control on dual-axis solar tracker," in Proc. ICOIACT, 2018, pp. 784–788.
[15] D. A. Flores-Hernández et al., "Heuristic approach for solar tracking optimization," IEEE Access, vol. 7, pp. 52755–52768, 2019.
[16] A. Mellit and S. Kalogirou, “Artificial intelligence techniques for photovoltaic applications: A review,” *Progress in Energy and Combustion Science*, vol. 34, no. 5, pp. 574–632, 2008.

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. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15469

[17] T. M. Razykov et al., "Solar photovoltaic electricity: Current status and future prospects," *Solar Energy*, vol. 85, no. 8, pp. 1580–1608, 2011.

[18] J. A. Duffie and W. A. Beckman, *Solar Engineering of Thermal Processes*, 4th ed. Hoboken, NJ, USA: Wiley, 2013.

[19] H. Mousazadeh et al., "A review of principle and sun-tracking methods for maximizing solar systems output," *Renewable and Sustainable Energy Reviews*, vol. 13, no. 8, pp. 1800–1818, 2009.

[20] S. Mekhilef, R. Saidur, and A. Safari, "Comparative study of different MPPT methods for PV systems," *Solar Energy*, vol. 91, pp. 1–16, 2013.

[21] N. K. Sharma, P. K. Tiwari, and Y. R. Sood, "Solar energy in India: Strategies, policies, perspectives and future potential," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 1, pp. 933–941, 2012.

[22] A. Al-Mohamad, "Efficiency improvements of photovoltaic panels using tracking systems," *Applied Energy*, vol. 79, no. 3, pp. 345–354, 2004.

[23] M. Rizwan, M. Jamil, and D. P. Kothari, "Generalized neural network approach for global solar energy estimation," *IEEE Transactions on Sustainable Energy*, vol. 2, no. 3, pp. 319–327, 2011.

[24] S. K. Saha and D. K. Saha, "IoT-based smart energy monitoring system," *International Journal of Energy Research*, vol. 44, no. 5, pp. 3562–3575, 2020.

[25] A. K. Saxena and V. Dutta, "Performance analysis of solar tracking system using microcontroller," *International Journal of Engineering Science and Technology*, vol. 2, no. 9, pp. 4253–4258, 2010.