International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 12 | Dec 2025 www.irjet.net p-ISSN: 2395-0072

Solar Power Management and SoC Prediction System Using Machine Learning

Ansh Shinde[1], Neeraj Maggavi[1], Akshay Siddannavar[1], Siddarood Naragund[1]

[1]Student, Dept. of Electronics & Communication, S.G.Balekundri Institute of Technology, Karnataka, India

---

Abstract - *Effective management of energy storage is crucial for off-grid solar systems. Traditional charge controllers depend on fixed voltage thresholds and often cannot predict critical battery drains before they happen. This paper introduces a smart solar management system that uses a Hybrid IoT Architecture, combining ThingSpeak for dataset creation and Blynk for real-time user monitoring. The system is designed on the ESP32 microcontroller and works with INA219, BME280, and BH1750 sensors. A unique Edge AI method is used where a Multiple Linear Regression (MLR) model, trained on historical data collected through ThingSpeak, is deployed directly on the ESP32 to predict the future State of Charge (SoC). This enables the system to manage loads (relays) proactively and inform users through the Blynk app and a local buzzer. Experimental results indicate that the predictive model accurately estimates battery backup time and significantly reduces unexpected power outages compared to traditional methods.*

Key Words: **IoT, Solar Energy, Machine Learning, SoC Prediction, ESP32, Battery Management, Remote Monitoring, INA219.**

1.INTRODUCTION

Solar energy is quickly becoming the best renewable energy source for off-grid use. However, the reliability of a solar photovoltaic (PV) system is mainly tied to its energy storage unit, the battery. In rural and off-grid settings, battery health often suffers because of inconsistent charging cycles caused by unpredictable weather, like cloudy skies or rain. A major challenge for these systems is the absence of smart energy management. Most traditional solar charge controllers available today work in a "reactive" way. They check the battery's terminal voltage and only cut off the load when the voltage drops below a set level, usually 11.5V or 11.7V. This approach often doesn't work well because voltage can change a lot with heavy loads. This can lead to early disconnections or, on the other hand, deep discharges that can seriously harm the battery. Additionally, these systems don't inform users about the actual "backup time" left, which can leave them unprepared for unexpected power outages. To fill these gaps, this paper introduces a Smart Solar Power Management System that uses Edge Artificial Intelligence (AI) and a Hybrid IoT Architecture. By combining the ESP32 microcontroller with a Multiple Linear Regression (MLR) model, the proposed system turns a regular solar setup into a smart device that can forecast future energy availability.

1.1 Motivation

The main reason for this project is the limitations of current "Voltage-Based" monitoring systems. Voltage is a delayed signal of battery capacity. By the time a typical controller notices low voltage, the battery might be significantly drained. Furthermore, standard controllers cannot see environmental factors. They do not consider how temperature or sunlight intensity impacts the charging rate.

There is a clear need for a system that can:

- Predict power failure before it happens.
- Enable users to remotely monitor and control their energy usage.
- Automatically reduce non-essential loads, like heavy sockets, to extend the runtime of critical appliances, like lights.

1.2 Objectives

The main goal of this work is to design a low-cost, predictive energy manager using the ESP32 NodeMCU. Unlike solutions that rely on the cloud, this system uses Edge AI, which means the decision-making algorithm runs locally on the microcontroller.

- Data Driven: We use ThingSpeak to log historical sensor data, including Voltage, Current, Lux, and Temperature, to train our prediction model.
- User Centric: We use the Blynk IoT App to provide real-time visualization and manual control over household loads.
- Predictive Logic: The system estimates the State of Charge (SoC) and actively manages the load relays to prevent blackouts and extend battery life.

2. PROPOSED SYSTEM DESIGN

The proposed system is built as a standalone, smart energy manager that runs on a "Sense-Predict-Act" cycle. Unlike passive data loggers, this system takes action to protect the battery using real-time and predicted data. The structure is divided into two main areas: the Hardware Layer and the IoT/Software Layer.

2.1 Hardware Architecture

The core processing unit is the ESP32 NodeMCU (38-pin version), powered by a dual-core Xtensa® 32-bit LX6 microprocessor. This specific controller was chosen for its high clock speed (240 MHz) and built-in Wi-Fi/Bluetooth stack. This allows it to run complex machine learning regression algorithms while keeping a stable connection to the IoT cloud.

The hardware is organized into three functional sub-systems:

1) Sensing Block (Data Acquisition) We achieve precise monitoring using the I2C communication protocol.
   a) Power Monitoring: We use two INA219 High-Side DC Current sensors:
      i) Sensor A (Address 0x40): This sensor is placed between the Battery and the Load to measure Discharge Current and Terminal Voltage.
      ii) Sensor B (Address 0x44): This sensor is placed between the Solar Panel and Charge Controller to measure Generation Current.
   b) Environmental Monitoring: A BH1750 digital light sensor measures solar irradiance (Lux) to assess "Solar Availability," while a BME280 sensor tracks ambient temperature, as battery capacity drops significantly at temperatures above 45°C.
2) Actuation Block (Load Management) To manage household appliances, we use a 4-Channel Relay Module (5V, Active Low). The relays are optically isolated to protect the microcontroller from back-EMF. The loads fall into two categories:
   a) Critical Loads: Lights and Fans (Connected to Relays 1 & 2)
   b) Non-Essential Loads: Heavy Sockets and Chargers (Connected to Relays 3 & 4), which we disconnect first during a "Low Power" event.
3) Feedback Block:
   a) Visual: A 0.96-inch OLED Display (SSD1306) connected via I2C gives on-site technicians real-time diagnostic data (Voltage, Current, Prediction).
   b) Auditory: A Piezo Buzzer produces distinct alarm sounds (e.g., 3 rapid beeps) when the AI predicts a critical battery failure within the next 2 hours.

2.2 Hybrid IoT Framework

To balance data logging requirements with real-time user control, a hybrid architecture was implemented:

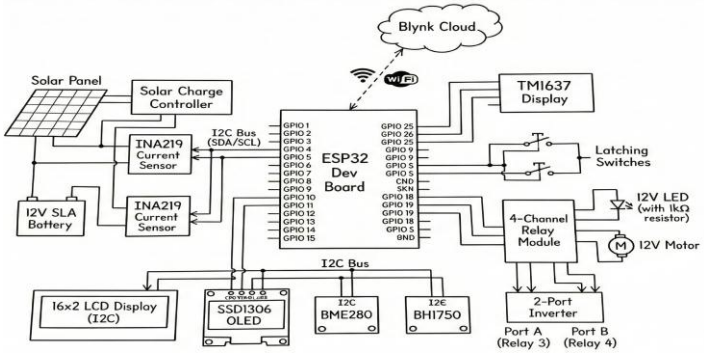

Figure 1: Block Diagram of Proposed System

- ThingSpeak (Cloud Layer): Serves as the database. The system uploads sensor telemetry every 15 seconds. This historical data is crucial for training the regression model.
- Blynk IoT (Edge Layer): Serves as the user interface. It utilizes Virtual Pins (V1 to V13) to receive real-time alerts and send control commands (switching relays ON/OFF) back to the ESP32 with low latency.

3. METHODOLOGY AND ALGORITHM

The intelligence of this system comes from its firmware. This allows the ESP32 to make independent decisions without needing a constant internet connection. The process consists of three stages: Data Logging, Training, and Edge Deployment.

3.1 Data Acquisition (Training Phase)

Before the system could predict battery life, we needed to understand how environmental factors affected battery drainage. We first set up the system in "Data Logging Mode." It uploaded sensor data to ThingSpeak every 15 seconds. The dataset included:

- Solar Generation Current (I_{sol}) and Voltage.
- Battery Voltage (V_{bat}) and Discharge Current.
- Ambient Temperature (T) and Light Intensity (Lux).

This data collection phase showed that battery efficiency drops significantly when the temperature goes beyond 40°C. Standard voltage-based controllers do not detect this pattern.

3.2 Edge AI Implementation (SoC Prediction)

To predict the State of Charge (SoC), we used a Multiple Linear Regression (MLR) model. This algorithm differs from simple threshold checks because it looks at several variables to predict the battery's future status. The regression

equation, based on our training data and embedded in the ESP32 firmware is:

$SoC_{pred} = (A.Lux) + (B.Temp) + (C.I_{sol}) + (D.V_{bat}) + K$

Where A, B, C, D are the trained weight coefficients and K is the intercept. The ESP32 calculates this prediction in real-time. If the model forecasts that the battery will reach a critical level <20% within the next hour, The system starts load shedding ahead of time.

3.3 System Operational Logic

The control flow, illustrated in Fig -2, operates as follows:

- Initialization: The system connects to Wi-Fi and calibrates the sensors.
- Sensing: The INA219, BME280, and BH1750 sensors collect real-time data.
- Prediction: The Edge AI model calculates the "Time to Empty."
- Decision Making:
  - Healthy State: If SoC is greater than 50%, all Relays are ON.
  - Warning State: If SoC is less than 40% or a high load is detected, the Buzzer beeps, and a notification is sent to the Blynk App.
  - Critical State: If the predicted SoC is less than 20%, Relays 3 and 4 (Sockets) are turned OFF to save power for lighting.

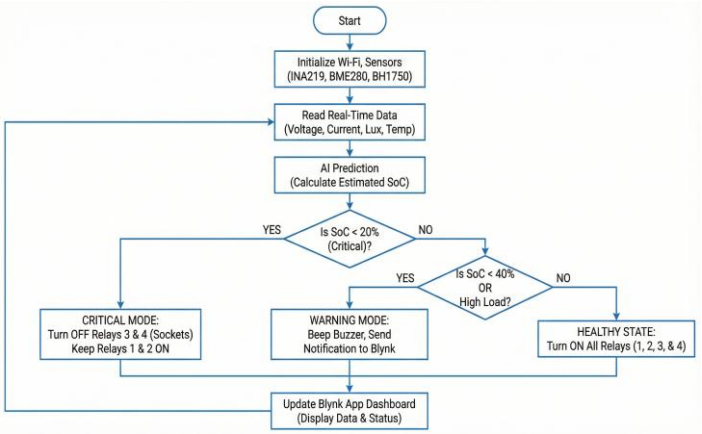

Figure 2: Flow Diagram of System

4. RESULTS AND DISCUSSION

The team tested the system under different environmental conditions to check the performance of the Edge AI model and the responsiveness of the IoT interface. The testing focused on how well the system could predict low-battery events and send remote alerts.

4.1 IoT Dashboard and Control

The user interface, created on the Blynk IoT platform, acts as the main control center. As shown in Fig 3, the dashboard offers real-time visibility into the system's parameters:

- Battery Gauge: Displays the live State of Charge (SoC) percentage (0-100%).
- History Graph**:** Monitors real-time current values of a complete day from the INA219 sensors.
- Status Indicators: Text widgets update dynamically based on the AI's decision (e.g., displaying "System Healthy" or "Critical: Unplug Sockets").
- Button Widges: To control relay switching from the Blynk application itself.

4.2 Performance Analysis

During the testing phase, the system underwent a forced discharge scenario. When the battery voltage dropped to 11.8V, the following automated actions were observed:

- Latency: The data transmission to the cloud showed a minimal latency of 1-2 seconds.
- Alerts: The system correctly triggered a push notification to the mobile device ("Low Battery Warning").
- Actuation: The "AI Suggestion" widget updated to display "Unplug Sockets," and the relay controlling the non-essential load was automatically disengaged, confirming the successful execution of the safety logic.

Figure 3: Blynk application

3. CONCLUSIONS

This paper describes the design and implementation of a Smart Solar Power Management System that uses the ESP32 microcontroller and Edge AI. By moving from a traditional voltage-based cutoff to a predictive data-driven model, the system effectively tackles the frequent problem of unexpected blackouts in off-grid setups.

The key achievements of this work include:

1. Intelligent Control: The embedded Multiple Linear Regression (MLR) model allows the system to foresee critical battery drain 1-2 hours in advance.
2. Hybrid IoT Architecture**:** The integration of ThingSpeak for data analysis and Blynk for real-time user interaction creates a robust monitoring ecosystem.
3. Battery Protection: By automatically shedding non-essential loads (sockets) while keeping critical loads (lights) on, the system extends the functional runtime of the battery during cloudy days.

Future enhancements will focus on implementing Deep Learning (LSTM) algorithms to integrate weather forecast APIs, allowing the system to adjust energy budgets based on tomorrow's predicted sunlight.

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude to our project guide, **Prof. Lohit Dalal**, **Assistant Professor**, Department of **Electronics & Communication**, for their valuable guidance, continuous encouragement, and technical support throughout the course of this work.

We also extend our sincere thanks to **Dr. Sridhar Iyer**, Head of the Department, and **Dr. B.R.Patagundi**, Principal of **S.G.Balekundri Institute of Technology**, for providing the necessary infrastructure and laboratory facilities to complete this project successfully.

Finally, we would like to thank our parents and friends for their moral support and constructive criticism which motivated us to work harder.

REFERENCES

[1] S. A. Sarkar, A. D. Nimbalkar, and A. S. Patil, "Design of Solar Panel Monitoring System Using ESP32 & IoT," in 2024 IEEE International Conference on Smart Power Control and Renewable Energy (ICSPCRE), 2024, pp. 1–6.

[2] G. A. Bhatane, V. J. Gond, S. V. Chaudhari, and P. William, "Real Time Performance Monitoring of Solar PV Panel IoT System for Energy Optimization," in 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM), Noida, India, 2024, pp. 1–6.

[3] S. Hossain, M. E. H. Sajib, and M. N. Hasan, "Design and Implementation of an IoT Based Solar Power Monitoring System," in 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2023, pp. 0211–0216.

[4] V. B. Gaikar, R. G. Deshmukh, and T. R. Kumar, "IoT based solar energy monitoring system," Materials Today: Proceedings, vol. 80, pp. 3535–3540, 2023.

[5] Texas Instruments, "INA219 Zero-Drift, Bidirectional Current/Power Monitor with I2C Interface," SBOS448G Datasheet, Aug. 2008, revised Dec. 2015.

[6] Espressif Systems, "ESP32 Series Datasheet," Version 4.3, Espressif Systems, Shanghai, China, 2023.

BIOGRAPHIES

Ansh Shinde
Final year B.E. student in ECE at **S.G.B.I.T college**. His interests include VLSI and Embedded Systems. He led the firmware development for this project.

Neeraj Maggavi
Final year B.E. student in ECE at **S.G.B.I.T college**. He specializes in Power Electronics and Circuit Design. He managed the hardware assembly and sensor calibration.

Akshay Siddannavar
Final year B.E. student in ECE at **S.G.B.I.T college**. He is enthusiastic about AI and Machine Learning. He developed the battery prediction algorithms for the system.

Siddarood Naragund
Final year B.E. student in ECE at . His focus is on Cloud Computing and Wireless Networks. He designed the Blynk IoT dashboard and server connections.