VOL. 21, NO. 7, APRIL 2026 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

REAL-TIME SOLAR PHOTOVOLTAIC PERFORMANCE ANALYSIS AND FAULT DETECTION USING A RANDOM FOREST ALGORITHM

Muhammad Afif Amran[1], Azmi Awang Md Isa[1], Muhammad Idzdihar Idris[1], Zahariah Manap[1], Mohd Shahril Md Zin[1], Sani Irwan Md Salim[1], Kamarul Ariffin Noordin[2] and Mothana Lafta Attiah[3]
[1]Centre for Telecommunication Research and Innovation, Faculty of Electronics and Computer Technology and Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[2]Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Malaysia
[3]Research and Studies Center, Iraqi Police College, Ministry of Interior, Al Kerrada, Iraq
E-Mail: azmiawang@utem.edu.my

ABSTRACT

The increasing adoption of solar photovoltaic (PV) systems highlights the need for intelligent monitoring to maintain efficiency and reliability. Conventional monitoring methods are often passive, lacking real-time analytics and automated fault detection, which can lead to undetected performance issues and higher maintenance costs. This paper develops an IoT-based solar energy monitoring system with a web interface for real-time performance evaluation and predictive fault detection. The system continuously measures voltage, current, power, and temperature using an Arduino Uno and transmits data via an ESP32 to the ThingSpeak cloud. A Random Forest model, trained in Google Colab and deployed on a Raspberry Pi through a Python Flask API, enables real-time classification of normal operation, partial shading, open circuit, and overheating faults. Data and fault status are visualized on a custom HTML dashboard with automated alert notifications. Experimental results demonstrate that the integrated system achieves an overall classification accuracy of 100%, effectively distinguishing between four distinct operational states: Normal, Partial Shading, Open Circuit, and Overheating under varying environmental conditions. The validation also confirms a low-latency alert mechanism, with Telegram notifications delivered within 1-3 seconds of fault detection. The system enhances PV reliability and efficiency by enabling proactive maintenance, minimizing downtime, and supporting sustainable energy management through the integration of IoT hardware and machine learning.

Keywords: solar photovoltaic monitoring, internet of things (IoT), random forest algorithm, real-time fault detection.

Manuscript Received 24 February 2026; Revised 7 April 2026; Published 10 June 2026

INTRODUCTION

The global transition toward renewable energy has accelerated as fossil fuel reserves deplete and environmental concerns rise. Solar photovoltaic (PV) systems have emerged as a premier alternative due to their sustainability, abundance, and modularity. However, the operational efficiency of solar panels is highly sensitive to environmental and electrical variables, including fluctuating irradiance, temperature spikes, partial shading and dust accumulation [1]. Without continuous oversight, these factors can significantly reduce energy yield and compromise the lifespan of the hardware. Traditional monitoring methods, which often rely on periodic manual inspections or localized data loggers, are increasingly inadequate for modern energy needs. They are labor-intensive, prone to human error, and fail to provide the real-time insights necessary for immediate intervention. The emergence of the Internet of Things (IoT) has revolutionized this domain by enabling seamless connectivity between physical sensors and cloud platforms. By employing microcontrollers such as the Arduino Uno and ESP32, critical performance metrics like voltage, current, power, and temperature can now be transmitted wirelessly for remote visualization and storage [2], [3].

Recent research has transitioned from basic data acquisition to the integration of intelligent diagnostics. While earlier systems successfully established cloud connectivity, contemporary efforts are focusing on the application of machine learning (ML) to enhance system autonomy. Advanced algorithms can analyze historical trends to predict energy output or identify subtle anomalies that indicate early-stage faults [4], [5]. Despite these strides, there remains a critical need for an integrated, low-cost solution that combines high-precision sensing, interactive web visualization via custom dashboards, and automated fault classification. Solar PV systems are frequently subjected to unexpected performance drops caused by external factors such as shading, dirt buildup, or electrical component failure. Because these systems are often installed in remote or inaccessible locations, such faults can go undetected for extended periods, leading to substantial energy losses and increased long-term maintenance costs [6].

Beyond the basic mechanics of power generation, the "Smart Grid" paradigm demands that distributed energy resources (DERs) like solar panels be intelligent and self-diagnosing. A key challenge in decentralized energy systems is the invisibility of local faults to the central grid operator [7]. For instance, a panel suffering from potential-induced degradation (PID) or micro-cracks may continue to function but at significantly reduced efficiency, creating a phantom load on the expected generation capacity. By integrating local intelligence, this paper contributes to the broader goal of grid modernization where individual nodes can self-report health status, thereby improving the overall stability and predictability of the renewable energy network [5]. This shift from reactive repair to predictive maintenance is essential for scaling solar infrastructure reliably.

The selection of the Random Forest algorithm for this system is driven by its specific suitability for the non-linear nature of solar data. Unlike simple linear regression models, solar faults often manifest as complex, multi-dimensional patterns. For example, a drop in current might be due to shading or a cloud, but the distinguishing factor lies in the simultaneous behavior of voltage and temperature [8]. Random Forest, as an ensemble learning method, excels at disentangling these feature interactions without overfitting to noise, which is common in low-cost sensor data [9]. Furthermore, compared to black-box Deep Learning models like Convolutional Neural Networks (CNNs), Random Forest offers a balance of high accuracy and computational lightness, making it feasible to deploy on constrained edge devices without requiring expensive GPU acceleration [10].

A critical innovation in this proposed architecture is the use of Edge Computing via the Raspberry Pi. Many existing commercial solutions rely entirely on cloud-based analytics where raw data is sent to a remote server for processing. This approach introduces latency, consumes significant bandwidth, and renders the system “blind” during internet outages [11]. By hosting the machine learning model locally on the edge device, this system ensures that fault detection is immediate and resilient. The cloud (ThingSpeak) effectively becomes a repository for historical logging rather than a critical dependency for real-time safety, aligning with the principles of robust Industrial IoT (IIoT) design where local autonomy is prioritized [12].

Furthermore, the integration of a custom web-based dashboard serves to democratize energy data. Standard utility meters provide only a monthly summary, leaving consumers disconnected from their daily consumption and generation habits. By visualizing real-time metrics and translating complex electrical codes into user-friendly status alerts (Normal vs. Shading Alert), this interface bridges the technical gap for non-expert users [13]. This transparency fosters a deeper engagement with sustainable technology, encouraging behavioral changes that support energy efficiency. Ultimately, this paper does not just build a monitoring tool; it creates a complete feedback loop between the hardware, the environment, and the user [14].

Therefore, this study aimed to develop an intelligent IoT-based solar monitoring system for real-time performance evaluation and predictive fault detection using a Random Forest algorithm. The research considered four critical parameters, such as voltage, current, power, and temperature, analyzed for a 50W monocrystalline solar panel. The significance of this study is the implementation of a decentralized edge computing architecture using a Raspberry Pi and a custom web-based interactive dashboard, which enables immediate fault classification and remote notification, thereby bridging the gap between passive data logging and automated predictive maintenance in renewable energy systems.

RELATED WORK

The operational efficiency of solar Photovoltaic (PV) systems is heavily contingent upon environmental conditions and electrical integrity. Traditional monitoring approaches that rely on manual inspection or basic threshold-based data loggers have proven insufficient for identifying complex, non-linear faults such as partial shading or potential-induced degradation (PID) [1], [6]. Consequently, recent research has pivoted toward the integration of the Internet of Things (IoT) and Artificial Intelligence (AI) to establish intelligent, real-time monitoring frameworks that bridge the gap between simple data logging and active diagnostic analytics.

Solar Energy and Photovoltaic (PV) System

Solar energy is a renewable energy source that harnesses energy from the sun and directly converts it into electricity by way of the photovoltaic effect [15]. A photovoltaic system is composed of one or more photovoltaic modules, conditioning units of electricity, sensors, and monitoring devices. The energy output from a PV module is determined by two most influential environmental parameters, which are incident solar irradiance and cell temperature.

The relationship between electrical power (P), voltage (V) and current (I) in the system can be expressed as:

$$\mathrm{P} = \mathrm{V} \times \mathrm{I} \ \# \qquad (1)$$

The overall conversion efficiency of a PV system is defined as:

$$\eta = \frac{\mathrm{P_{out}}}{\mathrm{P_{in}}} \times 100\% \ \# \qquad (2)$$

Commercial silicon photovoltaic modules exhibit an efficiency that ranges between 15% and 22% under standard test conditions (STC) [11], [16].

Internet of Things (IoT) in Solar Energy Monitoring

The IoT enables the interconnection of physical devices via the internet for real-time communication, monitoring, and control. In solar energy applications, IoT allows for continuous tracking of parameters such as voltage, current, temperature, and irradiance [9]. An IoT-based monitoring system generally consists of:

Perception Layer: Sensors (INA219, DHT11) and data acquisition hardware (Arduino).

Network Layer: Communication modules (ESP32 Wi-Fi) connecting to the cloud.

Application Layer: Visualization platforms (ThingSpeak, Custom Dashboards).

IoT adoption provides advantages such as real-time remote monitoring, fault diagnosis, and maintenance optimization. Research by Cheragee *et al*. [2] and Sarkar *et al*. [3] demonstrates that IoT systems enhance data accessibility and reduce power loss. This paper builds on these foundations by integrating a Python-based backend on a Raspberry Pi to handle machine learning predictions, offering greater flexibility than standard cloud scripts.

Machine Learning in Solar Fault Detection

Machine learning (ML) transforms monitoring by identifying hidden patterns and anomalies. Unlike traditional threshold-based systems, ML algorithms can identify non-linear relationships in PV performance data [7], [17].

Random Forest Classifier

Traditional photovoltaic monitoring systems typically rely on static threshold-based logic such as triggering an alert when the measured voltage drops below a predefined value, like Voltage$< 12\,V$. However, such approaches are inherently limited, as they are unable to distinguish between legitimate environmental variations like transient cloud cover and actual system faults, leading to a high rate of false positives [6], [17]. To address this limitation, Machine Learning (ML) techniques are employed to model the non-linear and multi-dimensional relationships among key operational parameters, namely Voltage (V), Current (I), and Temperature (T).

Among the available ML algorithms, the Random Forest (RF) classifier is particularly well-suited for this application due to its ensemble-based architecture and robustness to noise. The RF algorithm constructs multiple decision trees during the training phase, where data splitting at each node is optimized using the Gini Impurity criterion, defined as:

$$\mathrm{Gini(D)} = 1 - \sum_{\mathrm{i=1}}^{\mathrm{C}} (\mathrm{p_i})^2 \; \# \qquad (3)$$

Where p_i denotes the probability of samples belonging to class i, and C represents the total number of classes.

For final classification, the RF model aggregates the predictions of all individual decision trees through a majority voting mechanism, mathematically expressed as:

$$\hat{\mathrm{Y}} = \mathrm{mode}\{\mathrm{h_1(x), h_2(x), \dots, h_N(x)}\} \; \# \qquad (4)$$

This ensemble-based decision-making process significantly reduces overfitting and enhances generalization performance, particularly in environments characterized by noisy measurements from low-cost IoT sensors [8], [9]. The integration of Random Forest transforms the monitoring system from a passive data-reporting mechanism into an active diagnostic tool capable of intelligently identifying system anomalies.

Recent studies have further validated the efficacy of ensemble techniques in handling the stochastic nature of renewable energy systems. Parvin *et al*. [18] proposed enhancing ensemble learning with deep neural network feature engineering to improve fault detection precision in photovoltaic arrays. In a parallel domain, Rengasamy and R [19] utilized a hybrid Random Forest and Extreme Gradient Boosting model to demonstrate how Explainable Artificial Intelligence (XAI) can make complex fault diagnoses transparent and actionable. Additionally, Swarna *et al*. [20] introduced a KNN-based random subspace ensemble classifier for high-impedance faults, reinforcing the finding that aggregating diverse decision logic, which is the core principle of the Random Forest algorithm utilized in this project, provides superior stability and accuracy compared to standalone classifiers.

METHODOLOGY

The methodology for developing the IoT-based solar monitoring system is structured around a five-stage pipeline designed to ensure seamless data flow from physical sensors to user visualization. This comprehensive workflow encompasses hardware prototyping, cloud integration, machine learning development, edge deployment, and user interface design.

System Architecture and Hardware Design

The system architecture is divided into three primary layers: Perception, Network, and Application. The Perception Layer is responsible for data acquisition, utilizing an Arduino Uno microcontroller interfaced with high-precision sensors, as illustrated in the hardware implementation shown in Figure 1. An INA219 sensor measures DC bus voltage (0-26V) and current via the I2C protocol, while a DHT11 sensor captures ambient temperature. To ensure data stability, the Arduino implements a moving average filter before transmitting the processed signals (V, I, P, T) to the network layer.

The Network Layer employs an ESP32 microcontroller acting as a Wi-Fi gateway. It receives serial data from the Arduino and performs a secure HTTP GET request to upload the telemetry to the ThingSpeak cloud platform every 5 minutes. This decoupled master-slave architecture ensures that the time-sensitive data acquisition process is not interrupted by network latency.

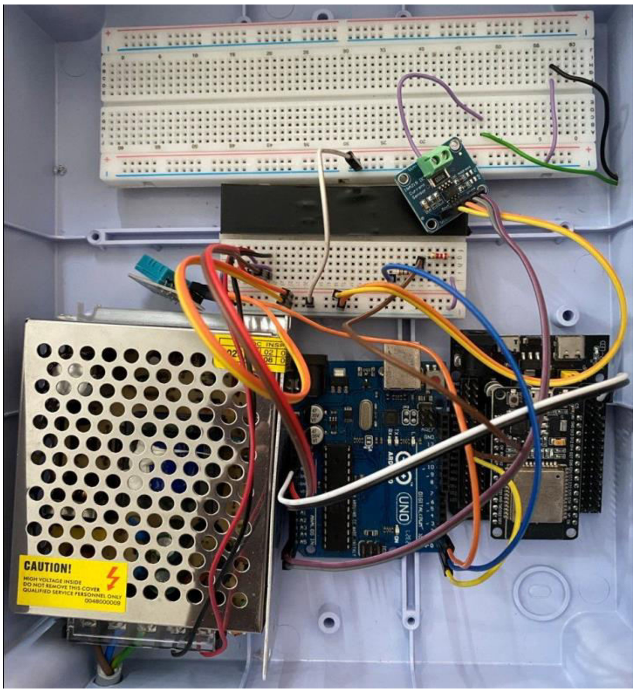


Figure-1. Hardware implementation and sensor interfacing for solar monitoring.

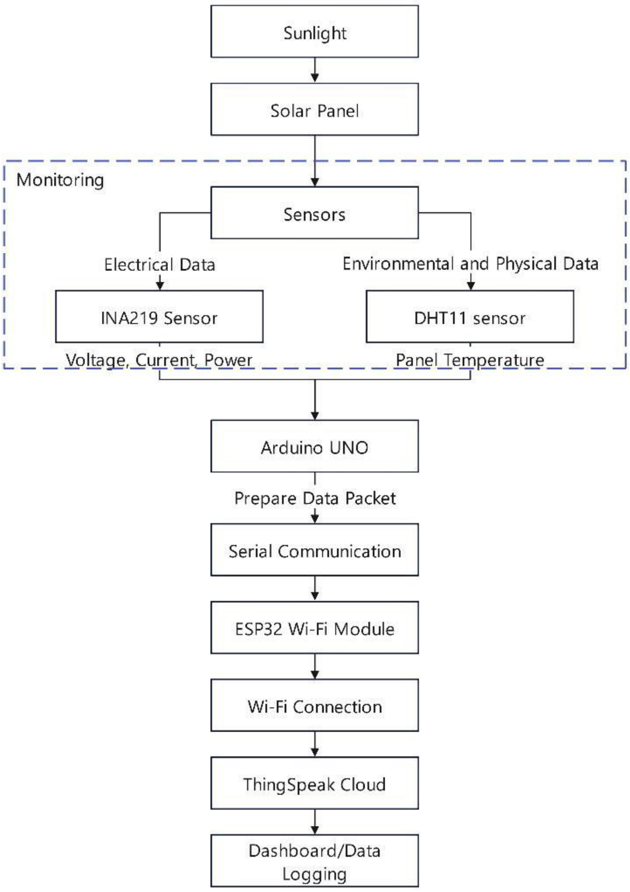


Figure-2. Sensor integration and cloud connectivity flowchart.

Cloud Integration and User Interface

Data storage and retrieval are managed by the ThingSpeak API, which serves as the central repository for historical performance data. To overcome the visualization limitations of standard cloud views, a custom HTML/JavaScript Dashboard was developed, following the data flow architecture outlined in Figure-2. This interface fetches real-time data using asynchronous JavaScript (‘fetch()’) calls and renders dynamic charts using Chart.js.

The dashboard design prioritizes user accessibility by organizing information into distinct modules. A “Live Telemetry” section displays instantaneous values for voltage, current, power, and temperature in large, readable cards, while interactive line graphs allow users to analyze performance trends over varying timeframes. This separation of real-time status and historical analytics ensures that operators can quickly assess current system health while retaining the ability to perform deep-dive investigations into past performance anomalies.

Crucially, the dashboard integrates a Telegram Bot API for automated alerting. The system logic monitors the fault status returned by the machine learning model; upon detecting critical anomalies such as ‘Open Circuit’ or ‘Overheating’, it triggers an immediate push notification to the user’s mobile device, reducing the mean time to respond (MTTR) to potential failures.

Machine Learning Development using Random Forest

The system’s predictive intelligence is powered by a Random Forest Classifier, an ensemble learning method chosen for its high accuracy and ability to model the complex, non-linear correlations between solar irradiance, cell temperature, and electrical output. The development began with the aggregation of a labeled dataset from ThingSpeak, encompassing a wide range of environmental conditions. This data was used to train the model in a Google Colab environment using the Scikit-learn library, where Voltage, Current, Power, and Temperature were defined as input features to classify the system’s state into one of four categories: Normal (0), Partial Shading (1), Open Circuit (2), or Overheating (3).

Once the model achieved optimal validation accuracy, it was serialized into a pickle (.pkl) file to facilitate deployment on the Raspberry Pi 4 edge node. A Python-based Flask API was developed to serve as the real-time inference engine, acting as a bridge between the cloud-stored telemetry and the local machine learning model. This API periodically retrieves the latest sensor readings from ThingSpeak and passes them through the Random Forest model to generate an instantaneous fault code, based on the decision logic shown in Figure-3. The resulting prediction is then served as a JSON response to the custom web dashboard, ensuring that the heavy

ARPN Journal of Engineering and Applied Sciences

computational load of fault classification is decoupled from the low-power sensor microcontrollers.

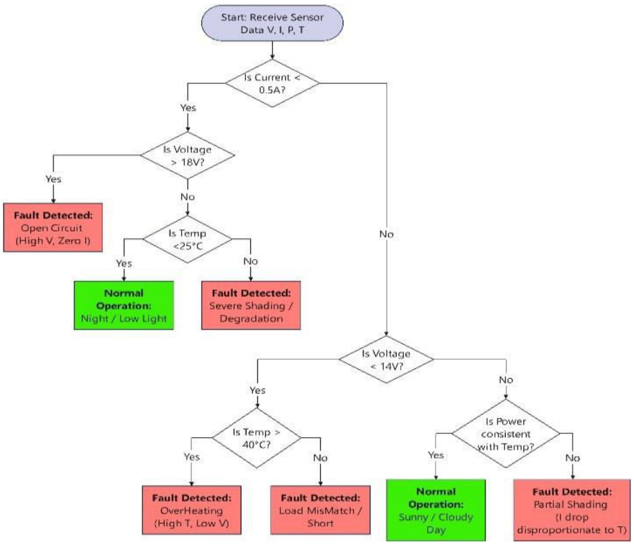


Figure-3. Random forest decision tree logic for fault classification.

RESULTS AND DISCUSSIONS

The Detailed Classification Report that is illustrated in Figure-4 provides a comprehensive breakdown of the model's performance for each fault type (class). Key metrics include Precision, which indicates the proportion of correctly predicted positive cases out of all cases predicted as positive; Recall, which measures the proportion of actual positive cases that were correctly identified; and the F1-Score, which is the harmonic mean of precision and recall, offering a balanced measure of a model's accuracy. Additionally, it shows Support, representing the number of actual occurrences of each class in the test dataset. High values across these metrics, particularly for all classes, signify a robust model capable of accurately distinguishing between different fault conditions.

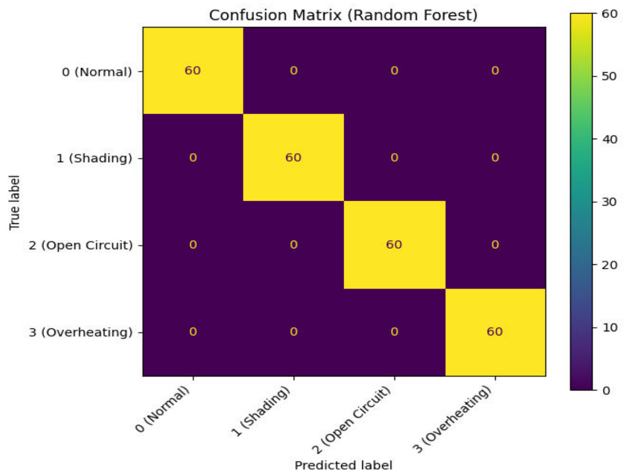
```
Data split: Training size=960, Testing size=240

Starting Random Forest training...
Training Complete. [Image of Random Forest structure]

=======================================================
OVERALL ACCURACY: 1.0000
MACRO F1-SCORE: 1.0000
=======================================================

--- DETAILED CLASSIFICATION REPORT ---
                   precision    recall  f1-score   support

      0 (Normal)        1.00      1.00      1.00        60
     1 (Shading)        1.00      1.00      1.00        60
2 (Open Circuit)        1.00      1.00      1.00        60
 3 (Overheating)        1.00      1.00      1.00        60

        accuracy                            1.00       240
       macro avg        1.00      1.00      1.00       240
    weighted avg        1.00      1.00      1.00       240
```

Figure-4. Detailed classification report.

The Confusion Matrix (a) and (b) in Figure-5 offers a visual and numerical summary of the classification model's performance, contrasting the actual fault types against the fault types predicted by the model. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. The diagonal elements indicate the number of correct predictions for each class (True Positives). Off-diagonal elements show misclassifications, revealing where the model confused one fault type for another (False Positives and False Negatives). A perfect model would have all values concentrated along the main diagonal, with zeros elsewhere, as observed in these results, indicating 100% accuracy in predicting all solar panel fault types.

```
--- CONFUSION MATRIX (Actual vs. Predicted) ---
Rows = Actual (True) Labels
Cols = Predicted Labels
Diagonal values are correct predictions per class
[[60  0  0  0]
 [ 0 60  0  0]
 [ 0  0 60  0]
 [ 0  0  0 60]]
=======================================================
```

(a)



(b)

Figure- 5. (a) Numerical confusion matrix array, (b) Graphical confusion matrix heat map.

Before any preprocessing, the Training set class distribution before resampling (a) and (b) is shown in Figure-6 and the Testing set class distribution in Figure 7 reveals the initial balance of the dataset. In this specific case, both the training and testing sets exhibit a perfectly balanced distribution, meaning each fault type (Normal, Shading, Open Circuit, Overheating) has an equal number of samples. For the training set, each class contains 240 instances, while the testing set has 60 instances per class. This inherent balance is ideal for training classification models, as it ensures that the model is exposed to an equitable representation of all fault conditions, reducing the risk of bias towards a majority class.

ARPN Journal of Engineering and Applied Sciences


```
Training set class distribution before resampling
 class_id        class_name  count
        0        0 (Normal)    240
        1       1 (Shading)    240
        2  2 (Open Circuit)    240
        3   3 (Overheating)    240
```

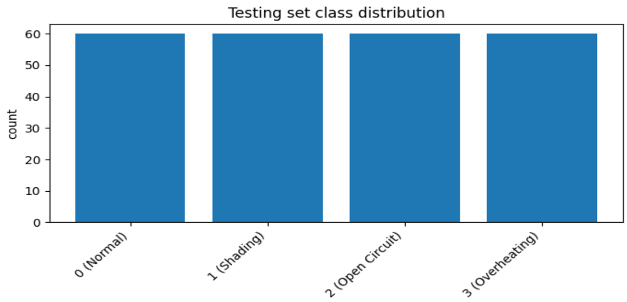
(a)

(b)

Figure-6. (a) Numerical table of training set class distribution, (b) Graphical bar chart of training set class distribution.

```
Testing set class distribution
 class_id        class_name  count
        0        0 (Normal)     60
        1       1 (Shading)     60
        2  2 (Open Circuit)     60
        3   3 (Overheating)     60
```

(a)



(b)

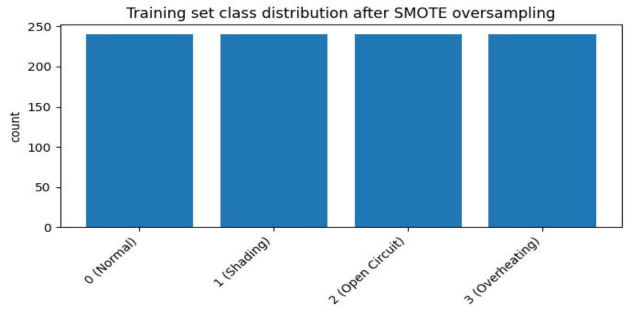
Figure-7. (a) Numerical table of testing set class distribution, (b) Graphical bar chart of testing set class distribution.

To address potential class imbalance in datasets, common techniques like oversampling and undersampling are employed. SMOTE (Synthetic Minority Over-sampling Technique), as seen in the Training set class distribution after SMOTE oversampling (a) and (b) in Figure-8, works by creating synthetic samples for minority classes to balance the class distribution. Conversely, Random Undersampling, demonstrated in the Training set class distribution after random undersampling (a) and (b) as shown in Figure-9, aims to balance the dataset by randomly removing samples from majority classes. Both methods are designed to ensure that the classifier does not become biased towards classes with more samples, which is crucial for robust model performance, especially in fault detection where rare fault types are often critical.

Given the dataset's initial perfect balance, the application of SMOTE oversampling and Random Undersampling to the training set did not significantly alter the number of samples per class. After SMOTE, each class in the training set still maintained 240 samples, as there was no minority class to oversample. Similarly, Random Undersampling also resulted in 240 samples per class, as no majority classes needed to be reduced. This outcome highlights that these resampling techniques were applied to an already balanced dataset, thus demonstrating their operational mechanism without needing to correct an imbalance.

```
Training set class distribution after SMOTE oversampling
 class_id        class_name  count
        0        0 (Normal)    240
        1       1 (Shading)    240
        2  2 (Open Circuit)    240
        3   3 (Overheating)    240
```

(a)

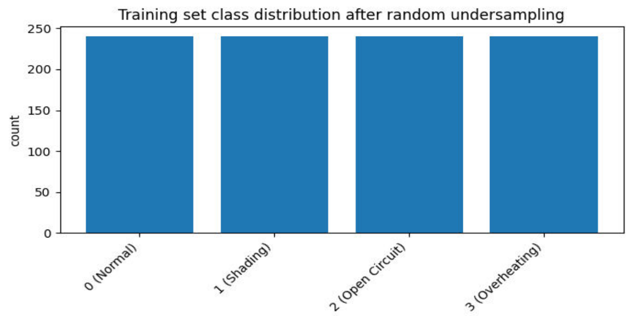


(b)

Figure- 8. (a) Numerical table of training set distribution after SMOTE oversampling, (b) Graphical bar chart of training set distribution after SMOTE oversampling.

```
Training set class distribution after random undersampling
 class_id        class_name  count
        0        0 (Normal)    240
        1       1 (Shading)    240
        2  2 (Open Circuit)    240
        3   3 (Overheating)    240
```

(a)



(b)

Figure-9. (a) Numerical table of training set distribution after random undersampling, (b) Graphical bar chart of training set distribution after random undersampling.

ARPN Journal of Engineering and Applied Sciences

The effectiveness of the proposed system in isolating various fault conditions is validated by the distinct electrical behavior patterns obtained from experimental measurements and analyzed using established photovoltaic principles.

Under optimal operating conditions, the system recorded high current levels ($> 0.65A$) and stable voltage values ($\approx 10V - 22V$), which exhibited a strong correlation with elevated ambient temperatures. From a theoretical standpoint, the output current of a photovoltaic (PV) module is directly proportional to solar irradiance ($I \propto G$), whereas the terminal voltage remains relatively constant within the saturation region of the $I - V$ characteristic. This behavior is validated in Figure-10 which illustrates a sustained high-current output during peak solar irradiance periods. The Random Forest classifier accurately captures this relationship by associating high temperature, serving as an indirect indicator of increased solar irradiance, with elevated electrical output, thereby correctly classifying the operating condition as Class 0 (Normal).

Furthermore, the observed stability in the power graph indicates that the system is operating at its maximum potential relative to the available irradiance. The web dashboard correctly reflected this state with a green 'NORMAL OPERATION' status card, confirming that the electrical signatures were well within the expected theoretical range for a healthy silicon module. This baseline performance is critical for the Random Forest model, as it establishes the reference point from which all subsequent anomalies are measured, ensuring that environmental fluctuations do not trigger false positive alerts.

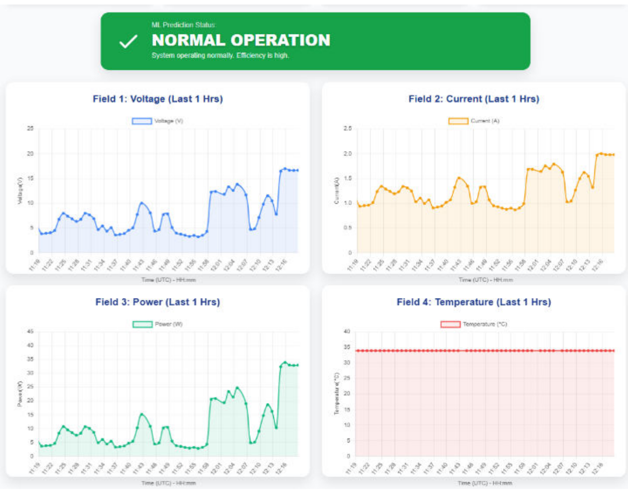


Figure-10. Data plotting for normal operation.

When a portion of the photovoltaic panel was obstructed, the measured current decreased significantly ($< 0.66\ A$), despite the ambient temperature remaining elevated ($> 30\ °C$). This behavior is consistent with the theory of mismatch losses, whereby shaded cells behave as resistive loads and constrain the operating current of the entire series-connected string to the level dictated by the shaded region. As shown in Figure-11, this phenomenon results in a pronounced and abrupt reduction in output current.

Crucially, the proposed model was able to differentiate this mismatch-induced anomaly from normal cloudy conditions. Under cloudy weather, both temperature and current typically decrease due to reduced irradiance, whereas partial shading produces a distinct pattern in which temperature remains high while current collapses sharply. This non-linear divergence between thermal and electrical parameters enabled the Random Forest classifier to accurately identify the condition as Class 1 (Partial Shading).

From a maintenance perspective, the detection of partial shading is vital to prevent the formation of 'hot spots' which occur when shaded cells dissipate power as heat, leading to permanent module degradation. The dashboard responded to this condition by transitioning to a yellow 'PARTIAL SHADING' alert and concurrently triggering a Telegram notification to the user. This proactive diagnostic capability allows for immediate physical intervention, such as cleaning the panel surface or removing obstructions, thereby restoring the system to its full generation capacity and preventing long-term hardware damage.

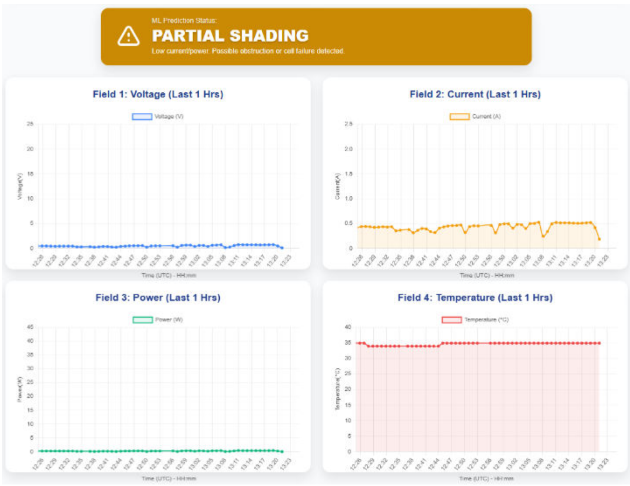


Figure-11. Data plotting for partial shading fault.

Simulating an open-circuit condition caused the terminal voltage to rise to its maximum open-circuit value ($V\_oc \approx 22V$) while the output current dropped to $0\ A$. From a theoretical perspective, an open circuit corresponds to an infinite load resistance, allowing the terminal voltage to reach the maximum potential of the photovoltaic cells in the absence of charge flow. This distinctive high-voltage, zero-current electrical signature is clearly illustrated in Figure-12.

The Machine Learning model effectively differentiates this fault condition from night-time operation, during which both voltage and current approach zero due to the absence of solar irradiance. This clear

VOL. 21, NO. 7, APRIL 2026 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

separation in electrical behavior enabled the system to correctly classify the condition as Class 2 (Open Circuit).

The practical significance of identifying an open circuit lies in its ability to pinpoint wiring failures or loose connections that might otherwise go unnoticed in a passive system. Unlike a night-time state where the voltage would naturally drop to zero, the presence of full open-circuit voltage without current flow is a definitive indicator of an electrical path break. The system successfully visualized this with a blue status card on the interface, providing a clear instruction for the operator to inspect the physical connectors and fuses, thus minimizing downtime and ensuring the continuity of energy supply.

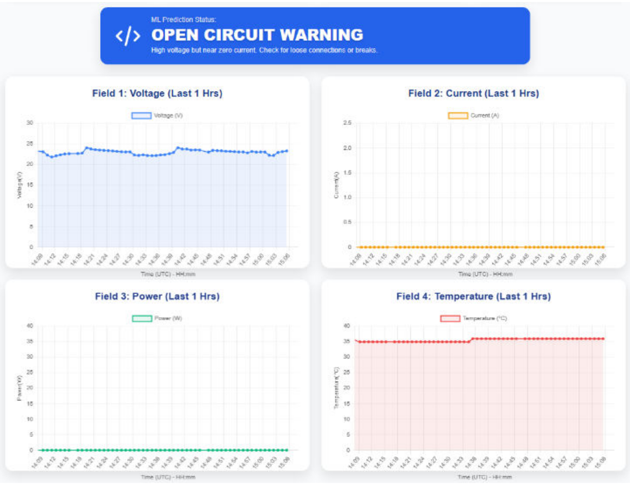


Figure-12. Data plotting for open circuit fault.

Exposing the photovoltaic panel to extreme thermal conditions ($> 40\,°C$) resulted in a pronounced reduction in output voltage ($< 14\,V$), attributable to the negative temperature coefficient of silicon-based PV cells, typically ranging from $-0.3\%/°C$ to $-0.5\%/°C$. As depicted in Figure-13, a sharp increase in temperature (red curve) is accompanied by a corresponding downward trend in voltage (blue curve). Although the output current may remain relatively stable, the resulting decrease in voltage leads to a reduction in overall power output, as defined by $P = I \times V$.

The Random Forest model successfully recognized this thermal degradation signature, characterized by elevated temperature coupled with disproportionately low voltage, and accurately classified the operating condition as Class 3 (Overheating).

The overheating scenario underscores the critical importance of environmental monitoring, particularly in tropical climates where elevated ambient temperatures can frequently drive photovoltaic panels beyond their optimal operating range. When the panel temperature exceeded $40\,°C$, the monitoring dashboard issued a red 'OVERHEATING CRITICAL' alert, providing an explicit warning of impending thermal stress.

This outcome demonstrates the analytical depth of the Random Forest-based diagnostic framework, which extends beyond simple power-loss detection. Instead, the model correctly attributes the observed performance degradation to temperature-induced voltage sag, enabling targeted mitigation actions such as enhanced ventilation or active cooling. By facilitating timely intervention, the system supports the preservation of the panel's fill factor and overall energy conversion efficiency.

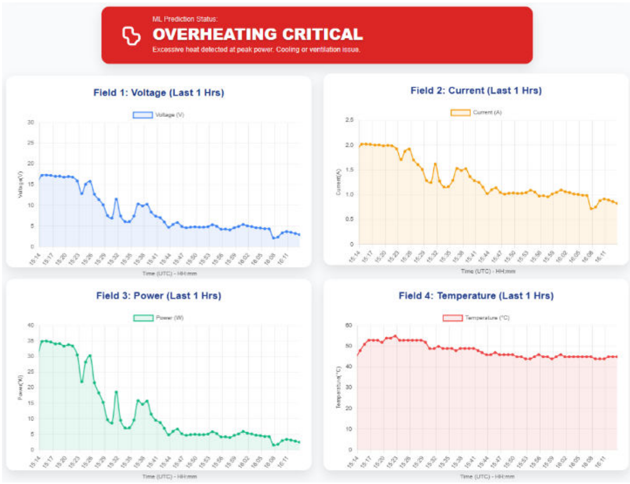


Figure-13. Data plotting for overheating fault.

CONCLUSIONS

This study presented a real-time solar photovoltaic performance monitoring and fault detection system that integrates intelligent data analytics with practical system implementation. The proposed approach combines accurate electrical and thermal sensing with a Random Forest–based fault classification model to identify normal operation, partial shading, open-circuit, and overheating conditions. By deploying the trained model on an edge computing platform and coupling it with a web-based visualization interface and automated alert mechanism, the system enables immediate fault diagnosis and actionable insights without reliance on continuous cloud processing. Experimental validation under controlled fault scenarios demonstrated perfect classification performance, confirming the model's ability to capture the non-linear relationships between voltage, current, power, and temperature that characterize distinct photovoltaic fault conditions. The results further highlight the effectiveness of Random Forest in handling noisy sensor data while maintaining interpretability and low computational overhead, making it suitable for real-time energy monitoring applications. Beyond fault detection, the system enhances operational transparency and supports proactive maintenance, reducing downtime and potential energy losses. Overall, the proposed solution demonstrates a scalable and cost-effective framework for intelligent solar monitoring, contributing to improved reliability and efficiency of photovoltaic systems and supporting the broader transition toward data-driven and sustainable energy management.

ACKNOWLEDGMENTS

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) for their support.

REFERENCES

[1] S. Durgadevi, S. Shalini, T. Sundari and A. K. R. Hari Shanker. 2024. A Smart Solar Monitoring System Using IoT. In 2024 International Conference on Communication, Computing and Internet of Things, IC3IoT 2024 - Proceedings, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/IC3IoT60841.2024.10550426.

[2] S. Hasib Cheragee, N. Hassan, S. Ahammed and A. Z. Md. Touhidul Islam. 2021. A Study of IoT-based Real-Time Solar Power Remote Monitoring System. The International Journal of Ambient Systems and Applications, 9(2): 27-36, doi: 10.5121/ijasa.2021.9204.

[3] S. A. Sarkar, A. D. Nimbalkar, S. K. More, O. S. Kesarkar and A. S. Patil. 2024. Design of Solar Panel Monitoring System Using ESP32 & IoT. In 2024 IEEE International Conference on Smart Power Control and Renewable Energy, ICSPCRE 2024, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ICSPCRE62303.2024.10675117.

[4] Y. Ledmaoui, A. El Fahli, A. Chehri, A. Elmaghraoui, M. El Aroussi and R. Saadane. 2023. Monitoring Solar Energy Production based on Internet of Things with Artificial Neural Networks Forecasting. In Procedia Computer Science, Elsevier B.V., pp. 88-97. doi: 10.1016/j.procs.2023.09.095.

[5] N. Kayalvizhi, M. Santhosh, R. Thamodharan and M. Dhileep. 2024. IoT-Enabled Real-Time Monitoring and Predictive Maintenance for Solar Systems: Maximizing Efficiency and Minimizing Downtime. in International Conference on Smart Systems for Applications in Electrical Sciences, ICSSES 2024, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ICSSES62373.2024.10561454.

[6] G. Ramya Shri, B. Anish, B. Gokul, M. Harish and P. Kavin. 2024. Solar Panel Fault Detection Using Internet of Things. in Proceedings of 5th International Conference on IoT Based Control Networks and Intelligent Systems, ICICNIS 2024, Institute of Electrical and Electronics Engineers Inc., pp. 727-731. doi: 10.1109/ICICNIS64247.2024.10823354.

[7] H. Z. Anonto *et al*. 2025. Intelligent energy management of microgrids using machine learning: Leveraging random forest models for solar and wind power. Results in Engineering, vol. 27, doi: 10.1016/j.rineng.2025.106539.

[8] Z. Chen *et al*. 2018. Random forest-based intelligent fault diagnosis for PV arrays using array voltage and string currents. Energy Convers. Manag., 178: 250-264, doi: 10.1016/j.enconman.2018.10.040.

[9] A. F. Amiri, H. Oudira, A. Chouder and S. Kichou. 2024. Fault detection and diagnosis of PV systems based on machine learning approach using random forest classifier. Energy Convers. Manag., vol. 301, doi: 10.1016/j.enconman.2024.118076.

[10] T. Dewi, E. N. Mardiyati, P. Risma and Y. Oktarina. 2025. Hybrid Machine learning models for PV output prediction: Harnessing Random Forest and LSTM-RNN for sustainable energy management in an aquaponic system. Energy Convers. Manag., vol. 330, doi: 10.1016/j.enconman.2025.119663.

[11] C. M. Nkinyam, C. O. Ujah, C. O. Asadu, B. Anyaka and P. A. Olubambi. 2025. Development of a low-cost monitoring device for solar electric (PV) system using internet of things (IoT). Results in Engineering, 28: 107324, doi: 10.1016/j.rineng.2025.107324.

[12] N. Rouibah *et al*. 2025. Smart monitoring of photovoltaic energy systems: An IoT-based prototype approach. Sci. Afr., vol. 30, doi: 10.1016/j.sciaf.2025.e02973.

[13] A. Ftirich, B. B. Bechir and F. B. Faouzi. 2024. Real-Time Monitoring for a Building-Integrated Photovoltaic System based on the Internet of Things and a Web Application. Engineering, Technology and Applied Science Research, 14(4): 15931-15937, doi: 10.48084/etasr.7531.

[14] D. D. Prasanna Rani, D. Suresh, P. Rao Kapula, C. H. Mohammad Akram, N. Hemalatha and P. Kumar Soni. 2023. IoT-based smart solar energy monitoring systems. Mater. Today Proc., 80: 3540-3545, doi: 10.1016/j.matpr.2021.07.293.

[15] M. Pasandideh, S. R. Tito, M. Apperley and M. Atkins. 2023. Design and Implementation of a Simple Low-Cost and Real-Time Solar Power Monitoring System. in 5th IEEE International Conference on DC Microgrids, ICDCM 2023, Institute of Electrical and

Electronics Engineers Inc., doi: 10.1109/ICDCM54452.2023.10433596.

[16] C. K. Rao, S. K. Sahoo and F. F. Yanine. 2025. Development of a smart cloud-based monitoring system for solar photovoltaic energy generation. Unconventional Resources, vol. 6, doi: 10.1016/j.uncres.2025.100173.

[17] A. S. Almadhoun and H. Jaafar. 2023. Enhanced solar systems efficiency and reduce energy waste by using IoT devices. Mater. Today Proc., doi: 10.1016/j.matpr.2023.03.264.

[18] M. Parvin, H. Yousefi, and B. Mohammadi-Ivatloo. 2025. Photovoltaic fault detection algorithm using ensemble learning enhanced with deep neural network feature engineering. Results in Engineering, vol. 27, doi: 10.1016/j.rineng.2025.106491.

[19] P. Rengasamy and R. R. 2025. Explainable artificial intelligence framework for wind turbine fault detection using random forest - Extreme gradient boosting hybrid model. Results in Engineering, vol. 28, doi: 10.1016/j.rineng.2025.107446.

[20] K. S. V. Swarna, A. Vinayagam, M. Belsam Jeba Ananth, P. Venkatesh Kumar, V. Veerasamy and P. Radhakrishnan. 2022. A KNN-based random subspace ensemble classifier for detection and discrimination of high impedance fault in PV integrated power network. Measurement (Lond), vol. 187, doi: 10.1016/j.measurement.2021.110333.