Smart Intelligent Monitoring and Maintenance Management of Photo-voltaic Systems

Marian Kingsley-Amaehule*, Roland Uhunmwangho*, Nkolika Nwazor*, and Kenneth E. Okedu**‡

*Department of Electrical and Electronic Engineering, University of Port Harcourt, 5325, Choba, Rivers State, Nigeria

**Department of Electrical and Communication Engineering, National University of Science and Technology, PC 111, Muscat, Sultanate of Oman

**Department of Electrical and Electronic Engineering, Nisantasi University, Istanbul 25370, Turkey

(mkingsleyamaehule@gmail.com, roland.uhunmwangho@uniport.edu.ng, nkolika.nwazor@uniport.edu.ng, okedukenneth@nu.edu.om or Kenneth.okedu@uniport.edu.ng)

‡ Corresponding Author; Kenneth E. Okedu, PC 111, Muscat, Sultanate of Oman

okedukenneth@nu.edu.om

Received: 04.12.2022 Accepted: 30.12.2022

Abstract- As the proliferation of solar photovoltaic (PV) system installation is on the rise, it is imperative to carry out new studies to monitor and optimize the maintenance management of solar PVs. The existing solutions of solar PVs monitoring and optimization are usually based on non-holistic approaches in solving the identified problem. In a bid to breach the research gap, this paper proposed and implemented a holistic model for solar PVs monitoring, maintenance, and management, considering Internet of Things (IoT). A mathematical model representing the solar PVs and the algorithms for its implementation were carried out, along with a designed embedded expert system as a proof of concept. Efforts were made to collect real-time data at both fog and cloud levels, in order to demonstrate the robustness of the control topology employed. The result analysis showed that the overall accuracy of the developed expert embedded system is 98.95%, which indicates that it can be used for effective and high reliability performance of solar PVs. Comparison of the sampled data collected at fog and cloud levels revealed that the system has 100% integrity in data communication, as well as 98% availability while simultaneously carrying out fault identification, classification and immediate analyses of the variables in real time. The knowledge gained in this research could be extended as future directions in other engineering fields for asset maintenance, management and artificial intelligence schemes.

Keywords Solar photovoltaic system, real time monitoring, four-level algorithm, fault identification, fog and cloud.

1. Introduction

Recently, the focus on renewable energy is on the rise as a result of global reduction in the use of fossil fuel-based energy resources [1]. Globally, solar and wind energy are the most economical and widely employed renewable energy sources [2]. One of the main reasons why solar energy is widely used is because of its non-polluting nature, thus, making it a clean and eco-friendly [3]. Photo voltaic (PV) cell is a major component that is used to harness solar energy. When connected with other components like battery, charge controller, ac inverters, it forms a PV system (PVS) which can be used to convert the sun's energy to alternating current and voltage signals [4]. The major challenge to wide implementation of solar PVs at a very large scale is the initial cost of installation [5]. Given the huge cost that is usually associated with the installation of large-scale solar PVs, maintenance management becomes a critical concern in the system.

In the literature, some authors have worked on monitoring and maintenance management of photovoltaic systems. A system for the collection of selected parameters of a solar plant for a period of five years was developed in Reference [6]. The data collected included energy, power, alternating current (ac) and direct current (dc) parameters of the solar plant. They were collected and stored at online data base by using Internet of Things (IoT) device to query the inverter. The main purpose of the work was to use the data to build a machine learning model, in a bid to understand the characteristics of solar power plants and give the predicted energy results. Wael C. et al., [7] worked on the performance of solar photovoltaic system, considering the approach of temperature effects and angle of inclination of solar PVs. In that study, the solar PV was investigated experimentally by placing it on two different types of soil at inclinations of 0° and 30°. Solar photovoltaic remote monitoring system using IoT was presented in [8]. The approach employed the monitored voltage, current and temperature of the solar PV cell and sent the data over the internet to be viewed anywhere on the globe.

In another study carried out in Reference [9], Felipe et al., proposed a novel method for faults detection in solar PVs using thermo-graphic camera embedded in an unmanned aerial vehicle. The method aimed at detecting hotspot faults on PV panels by the combination of thermography, Global Position System (GPS) positioning and Convolutional Neural Networks (CNN). An online smart solar PV monitoring system was suggested in [10], for the analyses of Voltage Current (VI) curves, in order to identify failures caused by technical or unintentional factors without interrupting the solar PVs during operation, thus, mitigating the maintenance costs involved in the process.

In addition, data collection from a solar plant for over five years was carried out and utilized to build a machine learning model in [11], in a bid to understand the characteristics of solar power plants and give the predicted energy results. In this work, deep learning technique was used for training while Keras and Tensor flow were employed for obtaining the results.

IoT-based approach for solar power consumption and monitoring that allows users to monitor and control their solar plants using their mobile phones was proposed in [12]. The study employed a design that first sends the data to the cloud before the mobile application gets this data from the cloud using an Application Programming Interface (API). More so, Reference [13] proposed a generic methodology that took solar and energy consumption patterns into account while evaluating the performance of solar batteries using Monte-Carlo simulation. Real data on Sydney's Central Business District (CBD) energy and solar generation were used to confirm the effectiveness of the proposed strategy. Again, in [14] fuzzy logic was employed in a novel way combined with artificial neural networks by joining genetic, particle swarm optimization, and imperialist competitive algorithms to attain a fast and optimal method for Maximum Power Point Tracking (MPPT).

In light of the above, it could be observed that a common feature exists in the above reviewed literature works; each concentrated on a particular component of the solar PV system on a separate basis. None of the literature works adopted a holistic approach to solar PV system monitoring and maintenance management.

Considering the requirement of National Renewable Energy Laboratory (NREL), the best recommended practices should be adopted in maintenance management of solar PVs [9]. The key idea as presented by Reference [9] is that system components of any solar PVs should be inspected and monitored regularly by maintenance personnel. Consequently, this study intends to breach this research gap and serving as a tool for renewable energy industries.

This paper presents a designed solar PV system that will aid in remote management of standalone solar PVs, while implementing real-time monitoring, classification, and analyses of faults within the system. Apart from developing a model for complete characterization and maintenance management of solar PVs, this work also demonstrated that Atmega microcontrollers can actually be connected to fog and cloud devices without using WIFI devices or telecommunication routers. This approach can be applied in other embedded systems applications within IoT ecosystems as well as other fields of engineering for asset maintenance and management.

2. Methodology

The proposed architecture for complete monitoring and maintenance management of a standalone solar PV system is shown in Fig. 1. The four major parameters of the solar PV system to be monitored include: the photovoltaic cell parameters, outputs of the charge controller, battery parameters as well as inverter output parameters. These parameters are basically functions of either voltage, current or frequency which has standard values which can be measured by a properly designed data acquisition node. The real time state of photo voltaic cell, charge controller and solar battery can be predicted using their output voltage, while the state of inverter can be predicted using its output frequency. To carry out the automatic data elicitation from the solar PV, data analysis, fault identification and reporting, an expert system is needed. The design of the expert system started from the mathematical description of the principle of operation of the entire solar photovoltaic system componenets with the help to sensors to the algorithm implementation.

The principle of operations of the entire solar photovoltaic systems are based on defined beachmark values for sensor reading as follows:

- Solar cells have output voltages that are rated at standard test condition (STC) temperature of 25ºC . So, at 25ºC, a good solar panel should have the rated value of output voltage.

- Charge Controller (CC) has rated input and output voltage. Once the rated input voltage is supplied to the CC, a good charge controller is expected to output the required voltage.

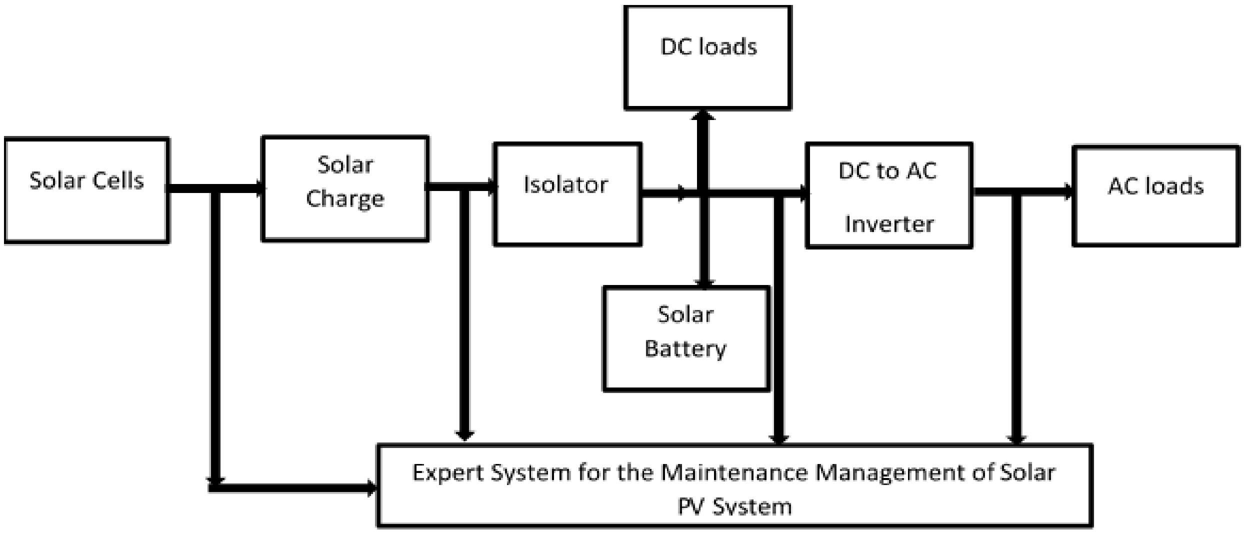

Fig. 1. Proposed architecture for solar PV monitoring and maintenance management

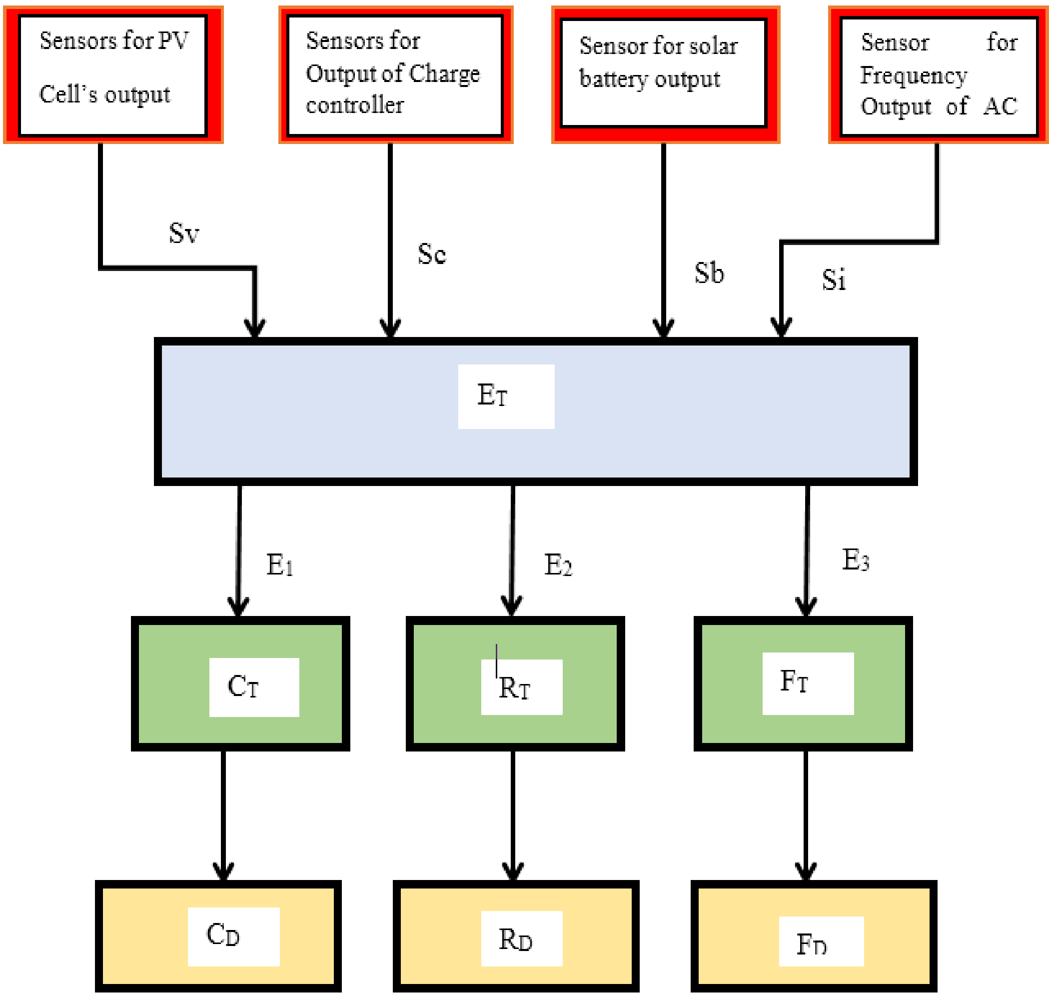

Fig. 2. Proposed model of embedded expert system for maintenance management of the solar PV system

- Solar Battery: Solar battery is characterised by amp-hour rating and depth of discharge (DOD). A test load can be used to check whether a battery is still in good condition or not. Also knowing and sensing the battery voltage can help one decide the state of the battery.

- Inverter: Every good inverter has base frequency of 50Hz or 60Hz. The approved standard of frequency being used in Nigeria is 50Hz. So, deviation from this benchmark shall be detected by Si.

2.1 Mathematical Model of the Proposed PVS Monitoring and Maintenance Systems

Consider the proposed model of the maintenance management of the solar PV system shown in Fig. 2.

Let

S_v = the sensors output from the solar panel,

T_v = the temperature of the solar panel and

V_v = the voltage output from the solar panel

The outputs expected from the solar panel are its temperature, voltage and current.

So primarily, S_v is a function of temperature *T*, and voltage *V*, since voltage is a function of current according to ohm's law [9]; this is represented mathematically in Equation (1)

$$S_v = f(T_v, V_v) \tag{1}$$

Similarly, the output of the charge controller is a function of the current and voltage, therefore:

$$S_c = f(V_c) \tag{2}$$

where,

S_c = sensor output from the charge controller

V_c = the voltage output from the charge controller

The sensor output for the battery will be,

$$S_b = f(V_b) \tag{3}$$

where,

S_b = sensor output from the solar battery

V_b = the voltage output from the solar battery

A faulty inverter usually has its frequency shifted from normal value which is 50 Hertz. It may also not have required output voltage. So, the output S_i expected from the inverter is

$$S_i = f(F_i, V_i) \tag{4}$$

Where,

S_i = sensor output from the ac inverter

F_i = frequency output from the ac inverter

V_i = voltage output from the ac inverter

Let $\oplus$ be a function that concatenate all the sensors output, *S*.

It implies then that

$$S = S_v \oplus S_c \oplus S_b \oplus S_i \tag{5}$$

Where,

S = concatenated output of the sensors

S_v = the sensors output from the solar panel

S_c = sensor output from the charge controller

S_b = sensor output from the solar battery

S_i = sensor output from the ac inverter

Similarly,

E, the output of the edge controller as shown in Fig.2 can be written as:

$$E = E_1 \oplus E_2 \oplus E_3 \tag{6}$$

E = the output of the Edge controller

E_1 = output 1 of the edge controller

E_2 = output 2 of the edge controller

E_3 = output 3 of the edge controller

Hence the edge controller transfer function E_T can be written as:

$$E_T = \frac{E}{S} = \frac{E_1 \oplus E_2 \oplus E_3}{S_v \oplus S_c \oplus S_b \oplus S_i} \tag{7}$$

Essentially, E_T is a digital transfer function comprising set of intelligent algorithms that take sensors outputs as their inputs, and after processing, brings out *E* as the expected outputs.

The output of the Edge *E* is in three categories are:

i. cloud input E_1

ii. Real-time monitoring input E_2, and

iii. Fog input, E_3.

It follows that the cloud function is given as

$$C_T = \frac{C_D}{E_1} \quad (8)$$

where,

C_T = cloud transfer function of the cloud gate way

C_D = cloud data

E_1 = output 1 of the edge controller which is also the cloud input

Similarly, the real time monitoring is given as

$$R_T = \frac{R_D}{E_2} \quad (9)$$

R_T = the real-time monitoring and reporting transfer function

R_D = Real-time data

E_2 = output 2 of the edge controller and also the Real-time and monitoring input.

And the fog Transfer function is given as

$$F_T = \frac{F_D}{E_3} \quad (10)$$

F_T = fog transfer function

F_D = fog data

E_3 = output 3 of the edge controller

Therefore,

$\boldsymbol{C_T}$ is a digital transfer function that specifies how the output E_1 of the edge controller E_T should be stored in the cloud.

$\boldsymbol{R_T}$ is also a digital transfer function that determines how exception data E_2 of the edge controller should be sent to the maintenance personnel.

$\boldsymbol{F_T}$ is the fog transfer function that determines how the output E_3 of the edge controller should be stored as fog data F_D.

2.2 Algorithms for the Software Development of the Expert System

Four algorithms are considered here and described in the following subsequent sections below:

- Data Elicitation Algorithm.
- Sensor Reading and Calibration Algorithm
- Fault Classification Algorithm
- Data Transmission to Remote Server Algorithm.

Algorithm 1: Data Elicitation Algorithm

This is used to extract information from the solar system expert.

Input: *Testbed physical addresses, pin numbers, data range, and data type*

Output: *mode classification of Testbed physical addresses, pin numbers, range of data*

SoftwareSerial gprsSerial(physical address of GPRS_modem);

Data_type solar panel_route_address = address _number;

Data_type charge controller_route_address = address _number;

Data_type battery_route_address = address _number;

Data_type battery_route_address = address _number;

Data_type inverter_route_address = address _number;

Data_type GPRS_route_address = address _number;;

Data_type solar _data_label and range;

Data_type chargecontroller_data_labe and rangel;

Data_type battery_data_label and range;

Data_type inverter_data_label and range;

void setup()

{gprsSerial.begin(data_baudrate);

pinMode(solar_route_address, state_mode);

pinMode(chargecontroller_route_address, state_mode);

pinMode(battery_route_address, state_mode);

pinMode(inverter_route_address state_mode);}

Algorithm II: Sensor Reading and Calibration Algorithm

Input: *Sensor Physical address*

Output: *sensor real data value*

For (sensor_address=0; sensor_address<4; sensor_address++)

*{Data_type sensor_value = analogRead(sensor_address)**;***

Data_type sensor_real_value **=** *(sensor_value* ***maximum ADC voltage)/adc_resolution;*

sensor real data value = map(sensor_value, R1, R2, M1, M2);

// M1 is the minimum rated value the sensor can read)

// M2 is the maximum rated value the sensor can read)

// R1 is the actual reading of the sensor in the absence of test signal)

// R2 is the actual reading of the sensor when subjected to the maximum concentration of the //test signal)}

Algorithm III: Fault Classification Algorithm

Input: *Data range of System default parameters*

Output: *Classified SMS fault alerts*

if (sensor data < default defined minimum range or > default defined maximum range)

{send classified expert defined fault;

Save data at both fog and cloud level;}

Else

{ save data at both fog and cloud level}

Algorithm IV: Data Transmission to Remote Server Algorithm

Input: *AT command of GPRS modem, APN, and API address* of the remote server

Output: concatenated data comprising solar (temperature and voltage), charge controller, battery and frequency levels

```
gprsSerial.println("set of AT_Commands for GPRS configuration");

gprsSerial.println("AT+CSTT=\"APN_address of GPRS modem\"");

gprsSerial.println("AT+CIPSTART=\"TCP\",\"
APN_address of remote server\",\"port_number of the server\"");

String str="GET
https://api.thingspeak.com/update?api_key=API_addres_of_
the_server&field1=" + String(solar_level)
+"&field2="+String(charge controller_level)
+"&field3="+String(battery_level)
+"&field4="+String(frequency_level);

Serial.println(str);

 gprsSerial.println(str);
```

2.3 Combined Flow Chart for the System Algorithm

Fig. 3 shows the high-level flow chart for implementation of algorithms I to IV. Here, we will supply the system with all the necessary default parameters needed for computations, decision making, and data sending which are the header files. Then collect real time solar panel, battery, charge controller and inverter present working parameters using sensors. Compare the collected real time system parameters with default parameters If there exist any meaningful variation, classify the fault or faults and appropriate SMS sent to the end-users of the information which are the maintenance personnel, vendors and owner of the asset before forwarding the system parameters to the fog computer and cloud server.

If there is no variation observed, send the collected data to the fog computer and cloud server and wait for acknowledgement after sending the data; if there is no acknowledgment after a specified time, report no acknowledgement received and continue.

2.4 System Design

Fig. 4 is the design that implements the model of Fig. 2. Equation (1) was implemented using sensor T and resistors R_1, R_2, and R_3. Equation (2) was implemented using resistors R_4, R_5, R_6, and R_7. Equation (3) was implemented using the relay, uln2003, R_8 and R_9. Equation (4) was implemented using the bridge rectifier circuit, R_{11}, R_{12}, the optocoupler (4N3J) and R_{10}. Atmega 328 microcontrollers were used to implement the algorithms of Section 2.2. Equations (8) and (10) shall be implemented using Wi-Fi module while Equation (9) shall be implemented using a GPRS module. The fog and cloud analysis shall be done Vis-a-Vis excel data base hosted at fog and cloud levels respectively.

Fig. 4 is categorised into input and output sections. The input sections are made up six sensors.

- The solar panel voltage sensor represented by R2 and R3
- The solar panel temperature sensor represented by R1 and thermistor.
- Charge controller input voltage sensor represented R4 and R5
- Charge controller output voltage sensor represented R6 and R7
- Battery voltage sensor represented by R8 and R9
- Inverter frequency sensor represented by R11, R12, bridge rectifier (D2, D3, D4, D5), opto-coupler 4N35 and resistor R10.

The output devices are made up of fog gateway, SMS gateway and cloud gateway. the SMS and cloud gate ways are implemented with GPRS sim900 module while the fog gateway is implemented with USB serial cable connected to a fog computer. The test load (LED in series with a resistor) is used to test the state of the battery while the edge controller contains algorithms that run the whole system.

2.5 Data Collection

ThingSpeak [15] as shown in Fig.5 was used to design an online database for storage and graphical representation of data collected from the edge controller [9]. The algorithms of Section 2.2 were developed using embedded C-language. Two sets of data were collected to test the validity of the proposed design. First, 100 samples of fog data were collected alongside the feedback data from the online server. This was done during the daytime and real time faults were also simulated in the system, and SMS messages received were noted. Next, an equivalent night data was also collected. Each set of data contains panel voltage and temperature, charge controller input and output voltage, battery voltage and inverter frequency as shown in Fig. 5.

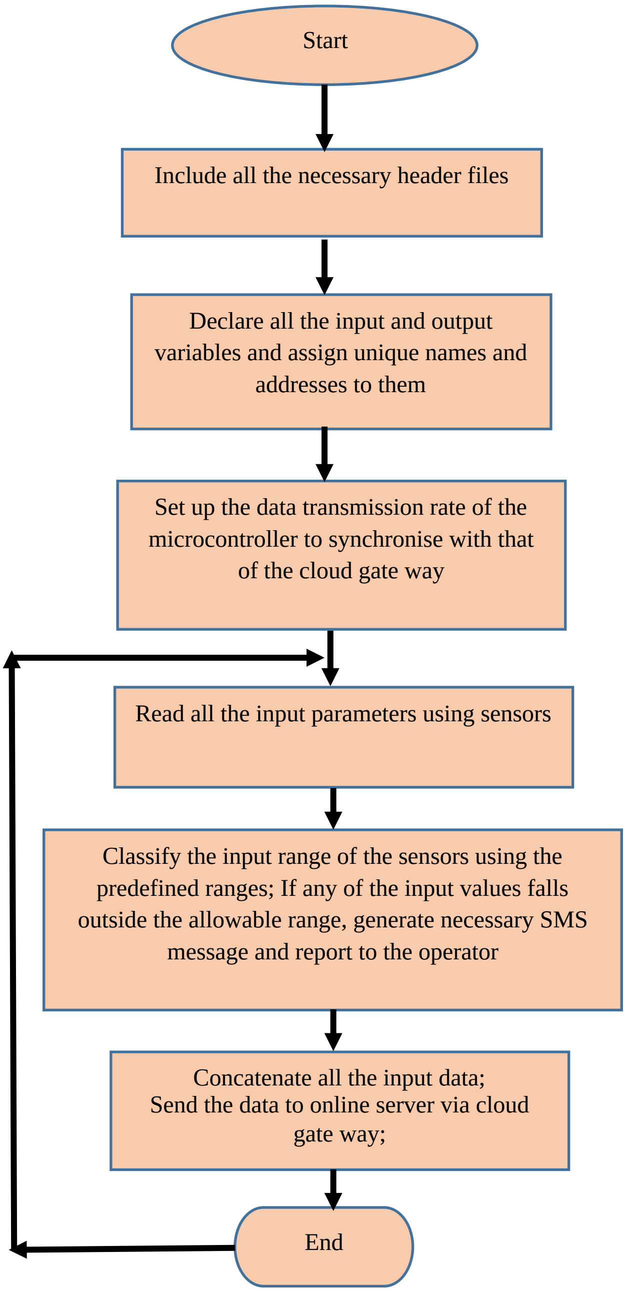

Fig. 3. High level flow chart for implementation of algorithms I to IV.

3. Results

The system design shown in Fig. 4 was implemented on printed circuit board as shown in Fig.6. R2, R4 and R6 of Fig. 4 were chosen to be 1000 Ohms while R3, R5 and R7 were chosen to be 10k variable resistor. IN4001 was used to implement D1 to D4. R11 and R12 were chosen to be 33000 Ohms each. 18 V 10 W solar panel, 10 Amp charge controller and 12V, 7 AH battery were used to implement the power source of the microcontroller system while 500 W 12 DC to 230 AC inverter was used to provide AC input to the system. DC to DC converter was used to provide stable 5 V DC voltage and adequate current to the General Packet Radio Service (GPRS) gateway modem. Thermistor interfaced to the microcontroller was used to measure the temperature of solar panel.

Fig. 7 shows the response of the system to the solar panel's temperature during the day while that of the night is shown in Fig. 8. The frequency response of the system is shown in Fig. 9.

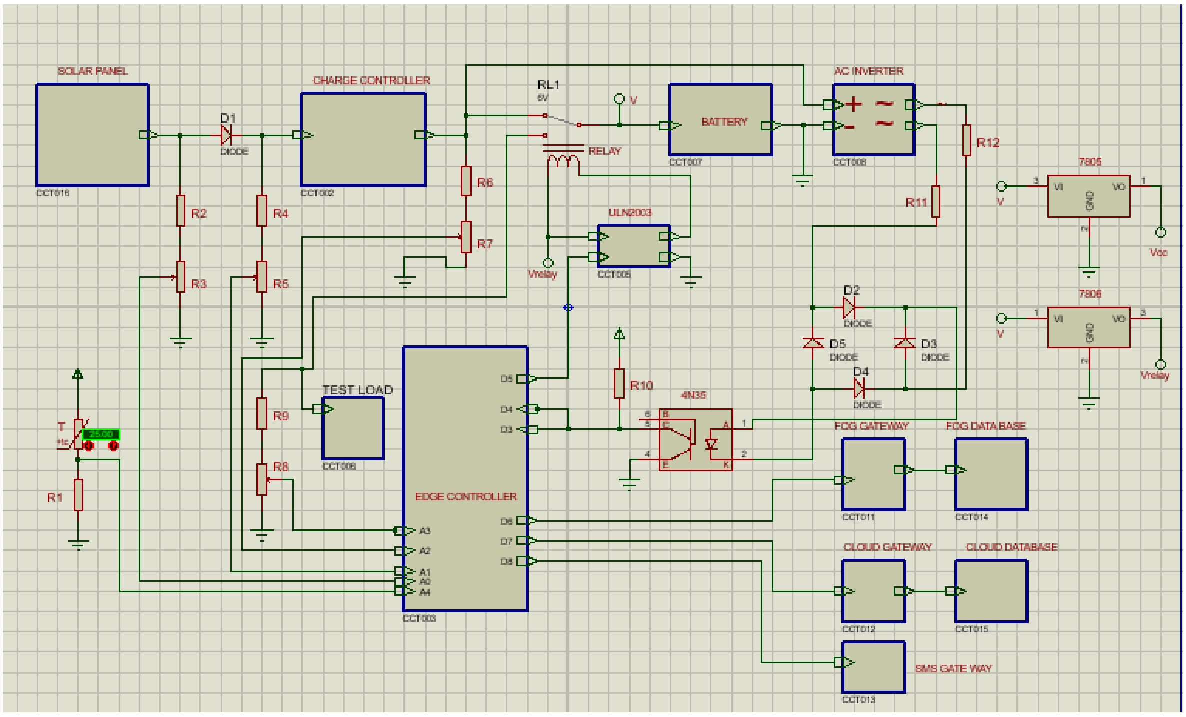

Fig. 4. System design

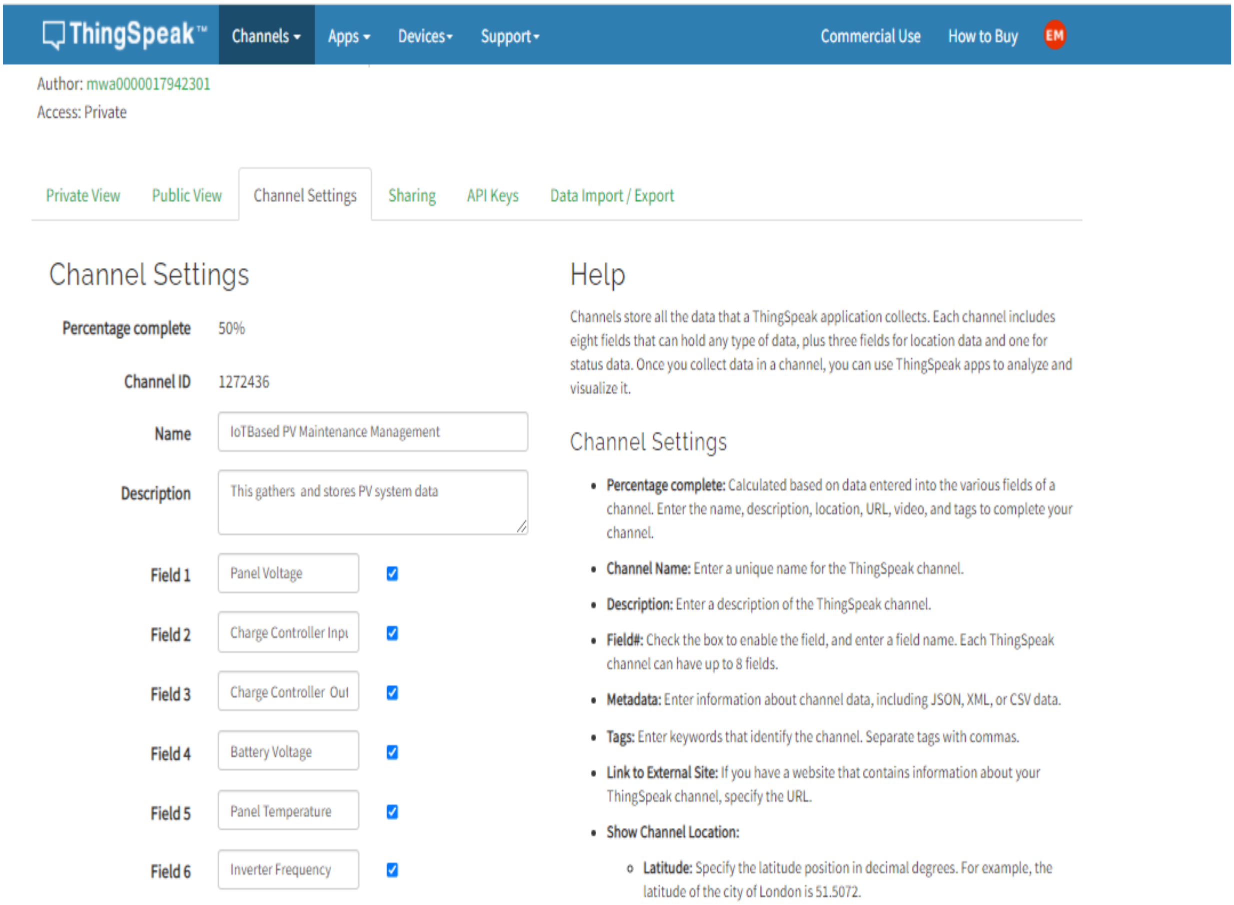

Fig. 5. ThingSpeak design for cloud data collection combined flow chart algorithm for the model

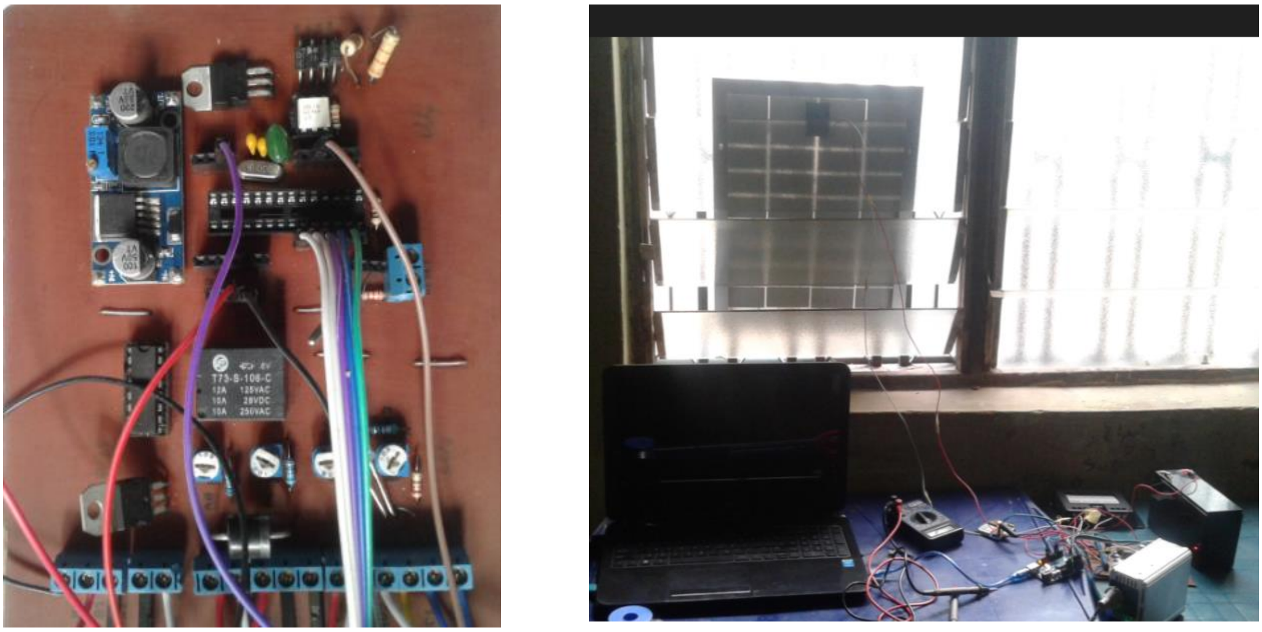

Fig. 6. Hardware implementation of the system design

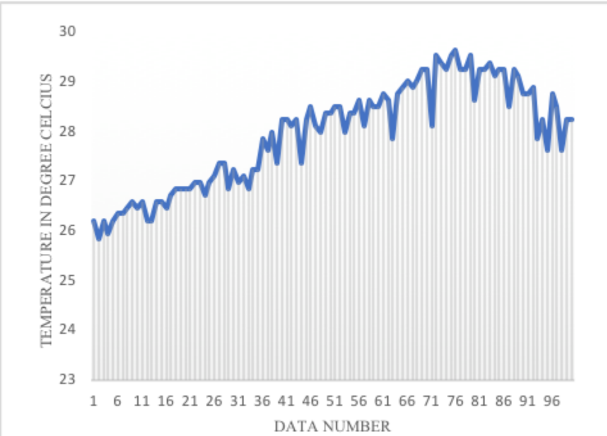

Fig. 7. System's response to solar panel temperature during the day

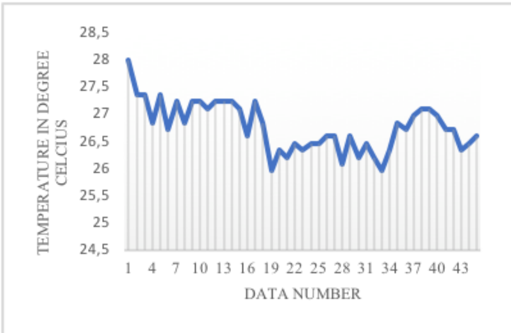

Fig. 8. System's response to solar panel temperature during the night

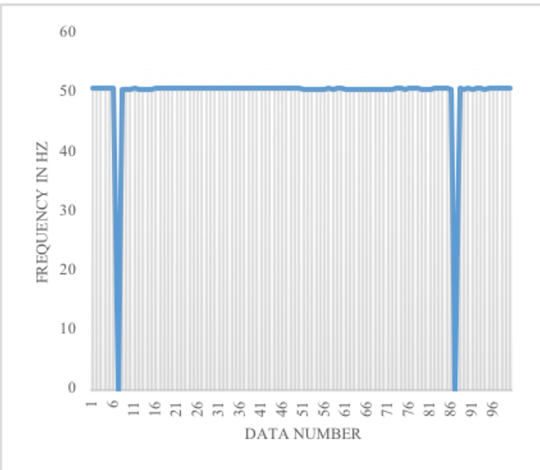

Fig. 9. Response of the system to the inverter frequency

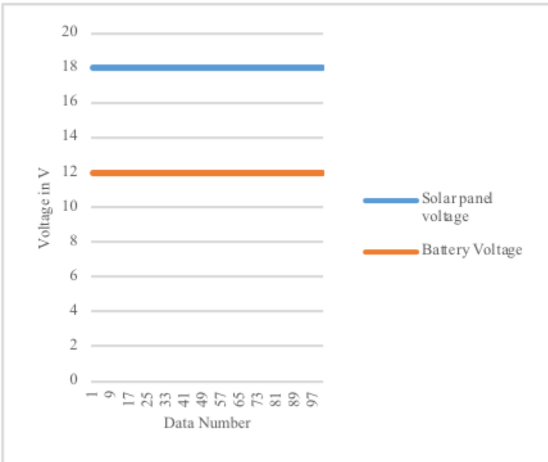

Fig. 10. Response of the system to both solar panel voltage and battery voltage

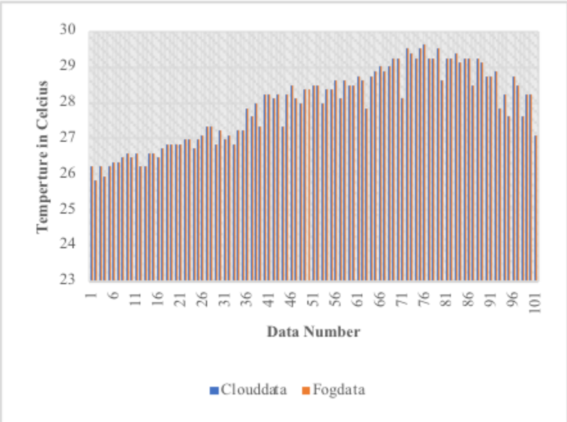

Fig. 11. Comparison of temperature data collected at fog and cloud levels

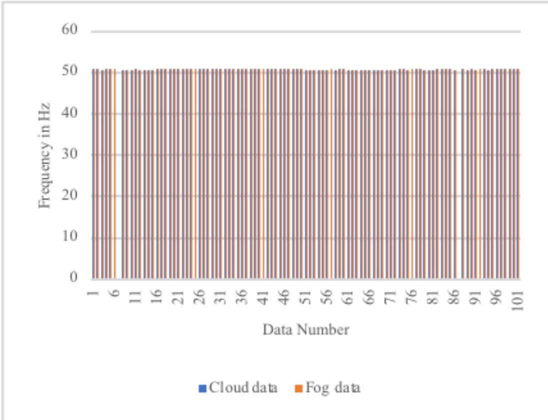

Fig. 12. Comparison of frequency data collected at fog and cloud levels

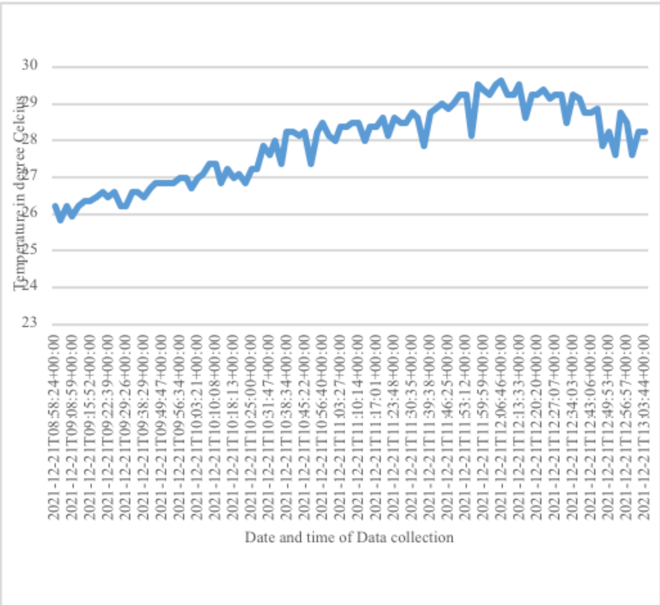

Fig. 13. The time stamp of the data collected at the online server

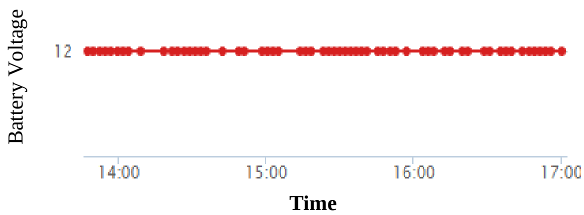

Fig. 14. The cloud analytic for the battery

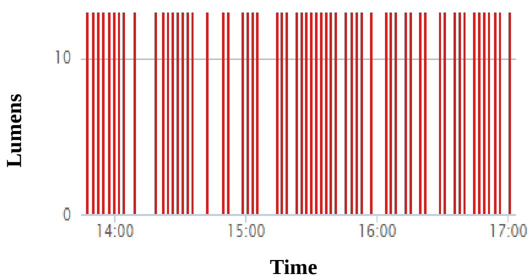

Fig. 15. The cloud analytic of the PV Panel voltage

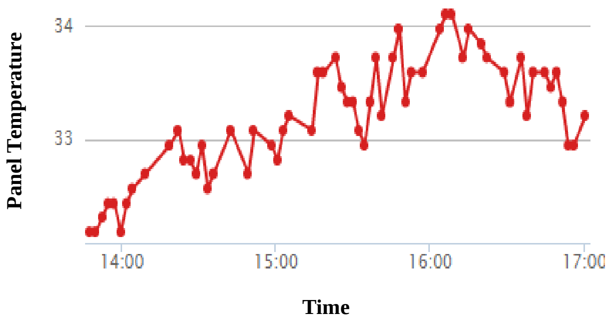

Fig. 16. The cloud analytic for PV temperature

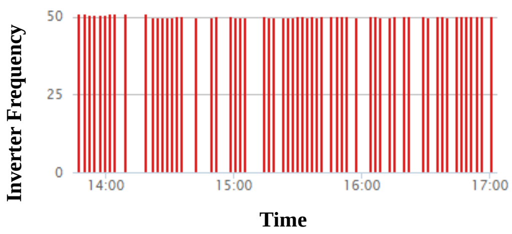

Fig. 17. The cloud analytic of the inverter frequency

Fig. 10 shows the response of the system to both at fog and cloud levels of the solar panel. Fig. 11 compares the temperature data collected both at fog and cloud levels. Figs. 14 to 17 represent the system analytic at cloud level.

4. Discussions

The test bed designed in this work accepted primarily three categories of inputs: voltage, temperature and frequency. Comparing Figs. 10 and 14, the system standard deviation of voltage measurement is zero. From Fig. 9 the average of the frequency response is 50.80 while that of temperature as computed from Fig.7 is 27.93. The average error in frequency measurement is 0.8. Therefore, accuracy of the system with respect to frequency measurement is $((50-0.8)/50) * 100 = 98.4\%$. The temperature of a standard thermometer reads 27.50 at the time of measurement. So, the system accuracy with respect to temperature is 98.44 %. With respect to voltage measurement the system accuracy is 100%. So, the average accuracy of the system is 98.95. This shows that the test bed designed for the solar PV system management is reliable. This is evidenced by the one-line graph obtained in comparisons of fog data and cloud data in Fig. 12.

Besides, it is seen from Figs. 11 and 12 that the developed system has 100% integrity as there is no single mismatch between the data received at the fog level and the data received remotely online. It means that the system can be used to monitor and manage the solar PV system remotely with high degree of reliability. Furthermore, In Figs. 9 and 12, the frequency dropped to zero when inverter fault was simulated. This occurs when we turned off the inverter it reads NAN (not a number) in the system but it is transmitted as zero which indicates that there is a fault in the inverter. A careful examination of Fig. 13 showed that the mean time between two sets of data sent to the server is 10 minutes. Implying that it will only take about 10 minutes for any system fault discovered at the edge level to be reported online. With SMS, this was noticed to be 20 seconds.

The solar PV cells usually have rated temperature and expected output voltage from manufacturers. Also, batteries usually have standard depth of discharge (DOD) which must not be exceeded while in operation. Frequency of inverters have standard value of 50 Hz. So by viewing the cloud analytics as shown in Figs. 14 to 17, one can actually know the state of the solar PVs per time.

Fig. 14 is the cloud analytic for the battery, while Fig.15 is that of the solar PV voltage. Fig. 16 shows the cloud analytic for solar PV temperature while Fig. 17 is that of the inverter's frequency. The responses of each of these figures represent the real time analysis of the selected parameter.

One of the salient parts of this study is that the complete six parameters of a typical solar PV was taken into consideration at the design level, for timely maintenance of solar PVs as compared to earlier existing solutions in the literature. In Reference [10], the presented work did data acquisition system for the solar PV system, however, the authors of the paper ended up designing a system that acquires data for solar panel alone. Also, Reference [10] designed a system that monitors solar PV cell and battery parameters, without considering the parameters employed in this current paper. Therefore, the target or unique feature of this research is that it presented a holistic approach to solar PV monitoring and maintenance while integrating IoT technology in the analytic.

5. Validation of the System Performance

A management system such as this is usually evaluated in terms of its availability, accuracy and integrity. The results of discussions so far showed that the system provided the same measurements as the physical measurement instruments. The system also has 100% integrity with respect to data transmission. During the period of data collection, it was observed from Fig. 9 that the system picked abnormal values of frequency twice in 100 samples. Thus, the system availability is 98%. It is therefore clear that a system like this can be deployed for monitoring and management of solar PV systems. As a case study, two independent stand-alone PV systems were set up with the following sizings: 18 W,12 DC solar panels, 30 Amp charge controller, 7.5 AH 12 V battery, 500 W, 220 AC inverter and 3 W AC energy saving bulb. Using solar calculation, each PV system is supposed to power the bulb for at least four hours every night. The model developed in this work was installed in one of the stand-alone solar PV systems. The two stand alone solar PV systems were operated for six months, while recording the Mean Time Between Failures (MTBF) for both systems. The results are shown in Table 1 and Fig. 18. Notifications were received from the monitored solar PV system, so proper maintenances were carried out within the period of study.

Table 1. MTBF for monitored and non-monitored solar PV systems

Month	MTBF with solar PV monitoring (hours)	MTBF without solar PV monitoring (hours)
January	648	360
February	624	288
March	672	240
April	360	312
May	480	216
June	408	264

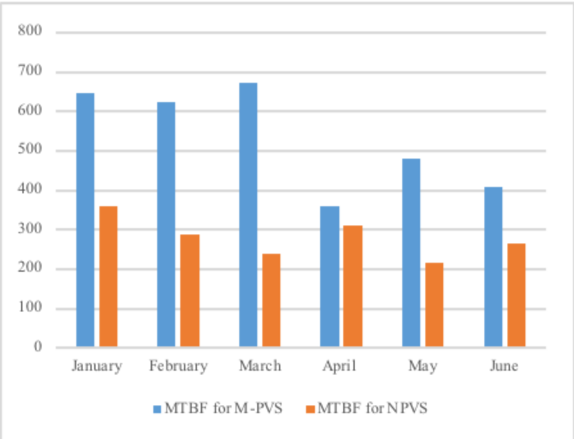

Fig. 18. Performance comparison of monitored and non-monitored PV system

From Table 1, the average MTBF for the monitored solar PV is 532 while that of the non-monitored solar PV is 280. It means that the model developed in this work improved the availability of the solar by a factor of 1.9, representing 90% improvement in the performance of the solar PV.

6. Conclusion and future directions

A smart model for total monitoring and maintenance of solar PVs was developed in this paper. A prototyped model was successfully implemented for easy replication of the study. The proof of concept (PoC) showed that the idea projected in this work is implementable. Complete six parameters of a typical solar PV were taken into consideration at the design level, for timely maintenance of solar PVs as compared to earlier existing solutions in the literature. The system availability of this study was 98%. It is therefore obvious that this smart model can be deployed for monitoring and management of solar PV systems. The idea presented in this work can be taken to commercial level by involving and engaging the necessary stakeholders Vis-a Vis meetings, publications, conferences among many other methods of disseminating research outputs.

References

[1] K. E. Okedu, A. Alghaithi, “Comparative Study of the Internal Dynamic Failures of Grid-connected Solar PVs: The case of Oman Power Network”, Frontiers in Energy Research-Smart Grids, vol. 10, no. 858803, pp.1-15 2022, DOI: 10.3389/fenrg.2022.858803.

[2] K. E. Okedu, and M. Al-Hashmi, “Assessment of the Cost of various Renewable Energy Systems to Provide Power for a Small Community: Case of Bukha, Oman”, International Journal of Smart Grid, vol.2, no. 3, pp. 172-182, 2018.

[3] S. M. Fahad, m. Abdallah, M. Ismali and M. E. Ahmed, “Review on Advancement in Solar Photovoltaic Monitoring Systems,” ISSN, no. 8, pp. 901-917, 2021.

[4] R. R. Vijaya, R. N V Ganapathi, S. V and P. Sugandha, “IOT based solar energy prophecy using RNN architecture,” in *E3S Web of Conferences*, India, 2020.

[5] K. E. Okedu, Husam AL Nadabi, Ahmed Aziz, “Prospects of Solar Energy in Oman: Case of Oil and Gas Industries”, International Journal of Smart Grid, vol.3, no. 3, pp. 138-151, September, 2019.

[6] A. Mohammadreza, M. K. Nallapaneni, E. Aref, A. Hamsa, K. V. d. O. Aline and S. C. Shauhrat, “Solar Pv Systems Design and Monitoring,” Photovoltaic Solar Energy Conversion-Technologies, Applications and Environmental Impacts, pp. 117-145, 2020.

[7] C. Wael, M. Chaabane, M. Hatem and B. Philippe, "Performance Evaluation of a Solar Photovoltaic System," *Energy Report*, pp. 400-406, 2018.

[8] K. Ankit and K. g. Suresh, "Solar Photovoltaic Remote Monitoring System Using IoT," in *RISE*, India, 2017.

[9] A. Harraiz, A. Marugan and F. P. Marquez, "Photovoltaic Plant Condition Monitoring using Thermal images Analysis by Convolutional Neutral Network-based Structure," *Renewable Energy*, vol. 153, pp. 334-348, 2020.

[10] A. T. Felipe, C. F. Melo and C. G. L. Freitas, "Design and Development of an Online Smart monitoring and Diagnosis system for Photovoltaic Distributed Generation," *Energies 2021, 14, 8552,* 2021.

[11] V. V. R. Raju, N. V. G. Raju, V. Shailaja and S. Padullaparti, "IOT based solar energy prophecy using RNN architecture," *E3S Web of Conferences*, vol. 184, p. 1007, 2020.

[12] M. Ali, "An IOT Based Approach for Monitoring Solar Power Consumption with AdaFruit Cloud," *International Journal of Engineering Applied Sciences and Technology,* pp. 335-341, 2020.

[13] M. Hassan, S. Saha and M. Haque, "A framework for the Performance Evaluation of Household Rooftop Solar," *International Journal of Electrical Power and Energy Systems,* pp. 1-18, 2021.

[14] F. Milad and P. Jafar Amiri, "Intelligent MPPT for photovoltaic panels using a novel fuzzy logic and artificial neural networks based on evolutionary algorithms," *Energy Reports,* pp. 1338-1348, 2021.

[15] ThingSpeak, "ThingSpeak for IoT Projects," 5 December 2019. [Online]. Available: www.thingspeak.com. [Accessed 6 January 2022].