



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

A Review on IoT-Based Solar Photovoltaic Monitoring Systems: Dust Effect, System Architectures and Performance Analysis

Deepak Sanjay Sapkal^a, Dr. Sunil Magan More^b, Krishna Vijay Gupta^c, Yashraj Hanumant Shelke^d, Lokesh Vikas Sapkar^e, Arya Deepak Korade^f, Mansi Santosh Karad^e

^{a,c,d,e,f} Student, Dept. of Electrical Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik, Maharashtra, India

^b Professor, Dept. of Electrical Engineering, Guru Gobind Singh College of Engineering and Research Centre, Nashik, Maharashtra, India

ABSTRACT

Solar photovoltaic (PV) energy is growing quickly, but how much electricity it actually produces depends on factors like dust, temperature, and sunlight, as well as how well it's maintained and operated. IoT technology makes it easier and cheaper to monitor PV systems from a distance and in real time. This paper looks at one main study about a smart solar PV monitoring system made using a Node MCU microcontroller and the Blynk app. The study measured how much power a 5.5 W solar panel lost over 18 days because of fly ash and normal dust. It also includes four other studies that cover IoT-SCADA system designs, fault detection in a greenhouse in a desert area, online tracking of maximum power points, and an IoT automation system used as a model. The paper compares different system designs, sensors, communication methods, cloud platforms, costs, and results using tables and charts. The findings show that fly ash caused an average loss of 49.16% in efficiency, while normal dust caused a loss of 17.46%. It also says that complete monitoring systems can be built for about 63 to 95 currency units. Open-source tools like Blynk, Thing Speak, Arduino Cloud, and Node-RED are good enough for small systems. Some challenges remain, like securing data, making systems work with more than one solar panel string, using Wi-Fi that only works a short distance, and not having predictive analytics.

Keywords: Solar photovoltaic, Internet of Things, Dust accumulation, Fly ash, NodeMCU, Blynk, SCADA, Fault detection, MPPT, Remote monitoring.

1. Introduction

Growing demand for clean energy has made solar photovoltaic (PV) systems one of the most widely adopted renewable technologies [1]. Power plants that burn fossil fuels emit CO, SO₂, NO_x and mercury, which contribute to acid rain and global warming [6]; PV generation avoids these emissions. However, the output of a PV panel is not constant. Elevated cell temperature reduces open-circuit voltage, humidity scatters light, and dust deposited on the glass blocks sunlight and lowers power output, especially in arid and dusty regions [1].

Without monitoring, such losses stay unnoticed and the useful life of a PV installation can be shortened. Commercial dataloggers are costly, need proprietary software and a constant power supply [1]. The Internet of Things (IoT) offers an alternative: low-cost microcontrollers, sensors and cloud platforms can collect voltage, current, power and environmental data and present them on a phone or web page anywhere.

1.1 Objective and scope of this review

This paper reviews the smart solar PV monitoring system of Pimpalkar et al. [1] and compares it with four reference works: the IoT-SCADA architecture with dual IoT platforms [2], the open-source SCADA with Banana Pi and Node-RED [3], the low-cost IoT monitoring with fault diagnosis for a greenhouse farm in an arid region [4], and the low-cost prototype for online monitoring of maximum output power [5]. The rooftop IoT automation system of Uzair et al. [6] is used as a model for organizing the review (introduction, literature review, architecture, methodology, results and conclusion).

1.2 Organization

Section 2 surveys the literature. Section 3 describes the generic architecture. Section 4 summarizes the methodology of the reviewed systems. Section 5 compares results through tables and charts, Section 6 lists challenges and future scope, and Section 7 concludes.

2. Literature review

2.1 Dust and soiling studies

Pimpalkar et al. [1] summarize several experimental studies on dust: an efficiency loss of up to 44 % for industrial dust in tropical regions, 50 % after six months without cleaning, 20 % for natural dust in Bangladesh with an Arduino-based wiping device, and 53 % power reduction at a 15° tilt angle. High relative humidity helps dust particles stick to the module surface, and smaller particles reduce performance more than large ones [1].

2.2 Low-cost dataloggers and IoT monitoring

Early monitoring systems relied on PC-based data acquisition cards and commercial software, which raised cost and limited use in remote areas [1]. Open-source Arduino dataloggers with 18-bit resolution and SD-card storage later complied with IEC accuracy requirements, and IoT versions added Wi-Fi and cloud access [1]. Pimpalkar et al. [1] follow this line using a NodeMCU and Blynk, adding a dust-effect study. Hamied et al. [4] and Rouibah et al. [5] use an Arduino Mega with an ESP8266 Wi-Fi module and a web page.

2.3 IoT-SCADA systems

SCADA architectures have evolved from monolithic to distributed, networked and IoT-based forms [2]. He et al. [3] designed an open-source SCADA using an ESP32-E as remote terminal unit (RTU), a Banana Pi M4 Berry as master terminal unit (MTU) with an MQTT broker, and Node-RED dashboards, at a total cost of CAD 94.5 and a running power of 3.19 W. He et al. [2] extended this with two IoT platforms (Arduino Cloud via MQTT and ThingSpeak via HTTP) and an ESP32-S3 camera for visual load inspection.

2.4 Fault detection and MPPT monitoring

Hamied et al. [4] integrated a simple fault-diagnosis algorithm into a low-cost microcontroller to detect disconnection, sand accumulation and short circuit in an off-grid system in the Sahara, and notified users by SMS. Rouibah et al. [5] implemented five maximum power point tracking (MPPT) algorithms (P&O, InCond, HC, VSS InCond and fuzzy logic) on an Arduino and monitored the maximum power online.

2.5 Automation of loads

Uzair et al. [6] note that most IoT-based PV systems only monitor and leave decisions to humans. Their rooftop system with five 125 W panels uses MQTT, a Raspberry Pi 4 and relays to switch loads automatically by a user-defined priority list. Simulation showed higher PV utilization than manual operation.

Table 1 - Summary of the reviewed works

Ref.	System / location	Controller	Platform	Key result
[1]	5.5 W panel, dust study, India	NodeMCU	Blynk	Fly ash loss 49.16 %, dust 17.46 %
[2]	PV + battery + load, Canada	ESP32-E, ESP32-S3	Arduino Cloud, Thing Speak	20-day run; load images
[3]	PV SCADA, lab, Canada	ESP32-E, Banana Pi	Node-RED (MQTT)	CAD 94.5; 3.19 W
[4]	100 W off-grid, Sahara	Arduino Mega, ESP8266	Web page, SMS	About 73 EUR; 13.5 Wh/day
[5]	120 W module, MPPT	Arduino Mega, ESP8266	Web page	63.04 EUR; FL 99.23 %
[6]	5 × 125 W panels, priority load switching	Arduino Mega, NodeMCU, RPi 4	MQTT broker	Utilization 75-95 %

3. Architecture of IoT-based PV monitoring

All reviewed systems follow a layered IoT architecture (Figure 1). The perception layer measures voltage, current, light (LDR), temperature, irradiance, dust or images. The network layer carries data through Wi-Fi using MQTT or HTTP. The middleware layer stores and processes data, and the application layer shows dashboards and alerts [2], [6].

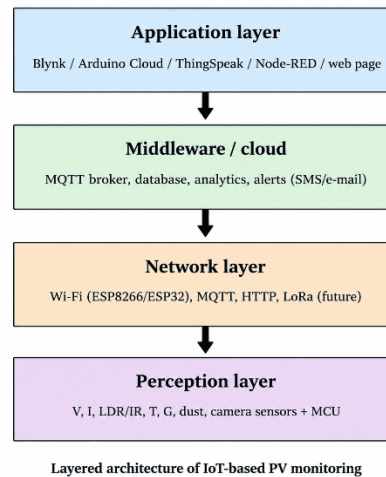


Fig. 1 - Layered architecture of IoT-based PV monitoring (prepared from [2], [6])

The first reviewed system [1] connects the solar panel, current sensor, voltage sensor, LDR and IR sensors to a NodeMCU, which sends data to the Blynk IoT platform and drives a DC motor that rotates the panel from east to west (Figure 2).

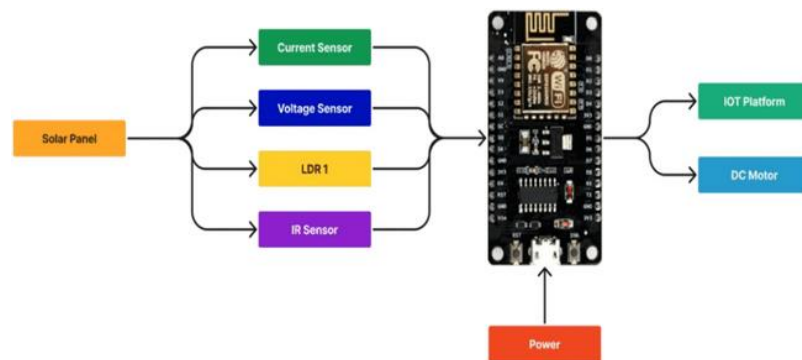


Fig. 2 - Block diagram of the smart solar PV monitoring system (adapted from [1])

Rouibah et al. [5] add a DC/DC boost converter and a PWM-controlled MPPT stage between the module and the load, with an Arduino board acquiring current, voltage, temperature and reference-cell data and sending them through an ESP8266 module to a web page (Figure 3).

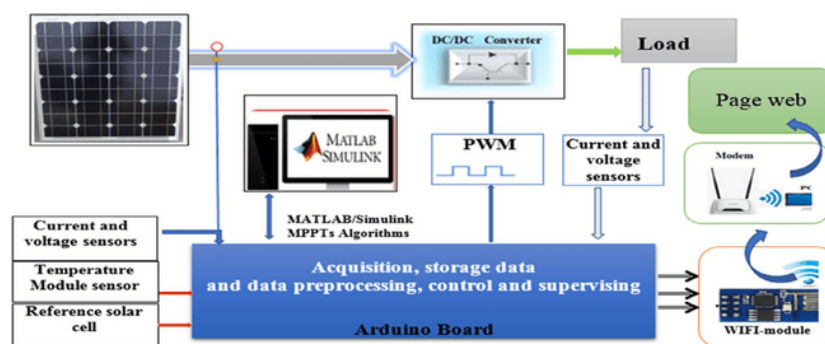


Fig. 3 - Block diagram of the IoT-based MPPT monitoring prototype (adapted from [5])

Table 2 - Components of the system in [1]

Sr.	Component	Sr.	Component
1	Solar panel (5.5 W)	7	IR sensor (b)
2	Current sensor	8	Breadboard
3	Voltage sensor	9	DC motor driver

Sr.	Component	Sr.	Component
4	LDR (a)	10	DC motor
5	LDR (b)	11	NodeMCU
6	IR sensor (a)	12	12 V 2 A adapter

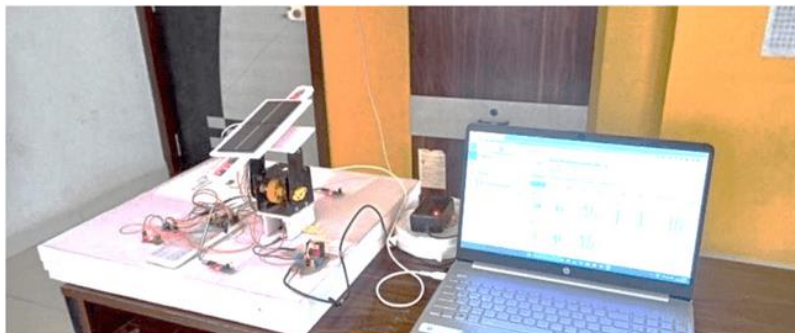
Table 3 - IoT platforms and protocols

Ref.	Platform	Protocol	Remarks
[1]	Blynk	Wi-Fi / cloud	Drag-and-drop app
[2]	Arduino Cloud + Thing Speak	MQTT, HTTP	Two platforms; redundancy
[3]	Node-RED on Banana Pi	MQTT (local)	Open source, no running cost
[4]	Own web page	Wi-Fi, SMS	Data not secured
[5]	Own web page (login)	Wi-Fi	XLS export of data
[6]	On-premises RPi 4	MQTT	Topic per sensor, 2 s

4. Methodology of the reviewed systems

4.1 Smart PV monitoring with dust study [1]

Sensors measure panel voltage, current and power; NodeMCU uploads readings to Blynk virtual pins V0, V1 and V2. If both LDR sensors detect sunlight the motor tracks the sun, otherwise the panel returns to the initial position. The experiment lasted 18 days: days 1 to 10 clean panel (baseline), days 11 to 14 fly ash, and days 15 to 18 normal dust, in quantities of 2, 4 and 6 g. Efficiency is calculated as actual power divided by maximum power (5.5 W) multiplied by 100 [1].

**Fig. 4 - Final experimental setup with live Blynk dashboard (from [1])**

4.2 SCADA-based systems [2], [3]

Two or three voltage sensors and three current sensors (ACS712) measure the PV module, battery and load. The ESP32-E publishes data by MQTT; a relay switches the load off when the battery voltage falls below 12.5 V [2]. In [3] the data go to a Banana Pi broker and Node-RED dashboard, and the system was tested during the total solar eclipse of 8 April 2024.

4.3 Fault diagnosis in an arid region [4]

A 100 W off-grid module supplies a small greenhouse with fan, lamp and water pump (Figure 5). An Arduino Mega measures PV current, voltage, irradiance and cell temperature; the fault algorithm runs on the microcontroller and users receive SMS alerts. Faults were created intentionally (shading, short circuit, open circuit and sand accumulation).



Fig. 5 - Greenhouse farm powered by the PV system in the Sahara (from [4])

4.5 MPPT monitoring [5]

The five MPPT algorithms were first simulated in MATLAB/Simulink and then coded on an Arduino Mega. The module current, voltage, load values, duty cycle, temperature and irradiance are uploaded to a password-protected web page.

5. Results and discussion

5.1 Effect of dust on efficiency [1]

The clean panel gave a maximum of 5.5 W at 11 V and 0.5 A. Averaged efficiency loss was 9.3 % for the clean panel, 49.16 % for fly ash and 17.46 % for normal dust. Figure 6 shows the day-wise loss and Figure 7 the effect of dust weight: 6 g of fly ash cut the power from 5.5 W to 2 W (64 % loss), while 6 g of normal dust reduced it to 4 W (20 % loss).

Table 4 - Efficiency loss of dusty panels [1]

Dust (g)	Fly ash P (W)	Fly ash loss (%)	Normal dust P (W)	Normal dust loss (%)
2	4.5	19	5.0	10
4	3.5	37	4.5	19
6	2.0	64	4.0	20
Average	3.33	49.16*	4.50	17.46*

* average over the experiment as reported in [1]

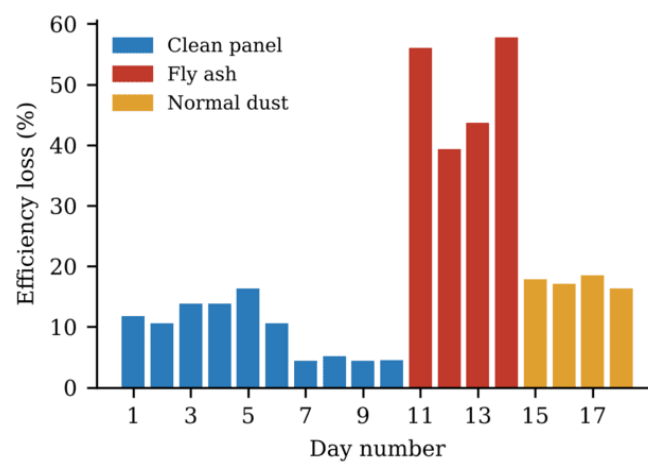


Fig. 6 - Day-wise efficiency loss for clean, fly ash and dust conditions (data from [1])

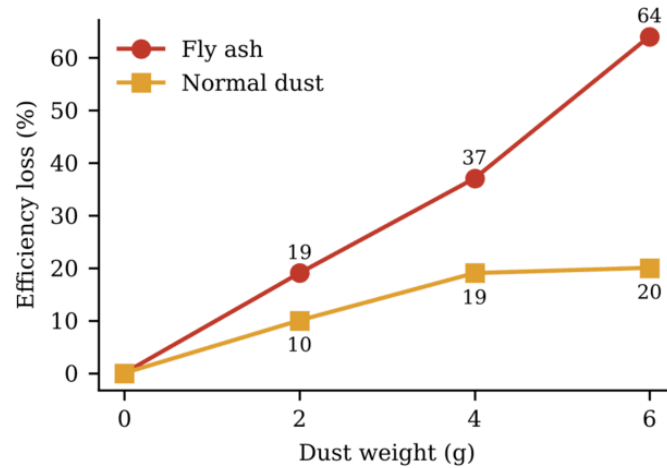


Fig. 7 - Efficiency loss versus dust weight (data from [1])

5.2 MPPT algorithms [5]

Under real conditions the fuzzy logic controller reached the highest efficiency of 99.23 % with a convergence time of 25 ms, while P&O was fastest (20 ms) but only 85.62 % efficient (Figure 8). The hill-climbing method was slowest (50 ms) and least efficient (83.58 %).

Table 5 - Comparison of MPPT algorithms [5]

Method	Power (W)	Tc (ms)	Eff. (%)	Complexity
P&O	32.65	20	85.62	Low
InCond	33.48	23	97.29	Low
HC	32.73	50	83.58	Low
VSS InCond	32.65	45	97.76	Medium
FL	33.54	25	99.23	Complex

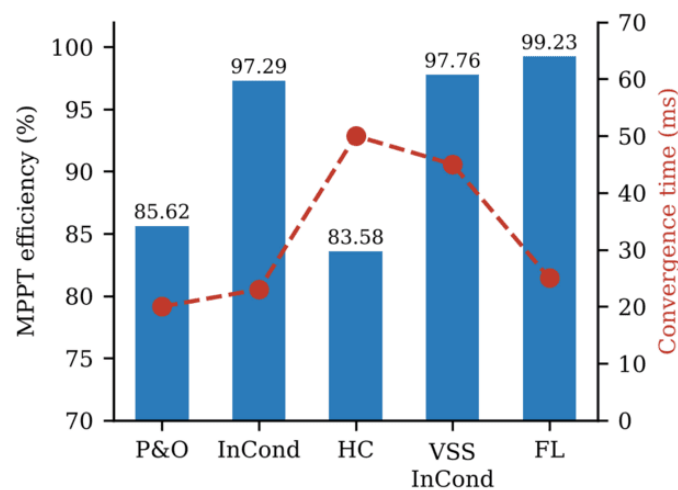


Fig. 8 - MPPT efficiency and convergence time (data from [5])

5.3 Cost and power consumption

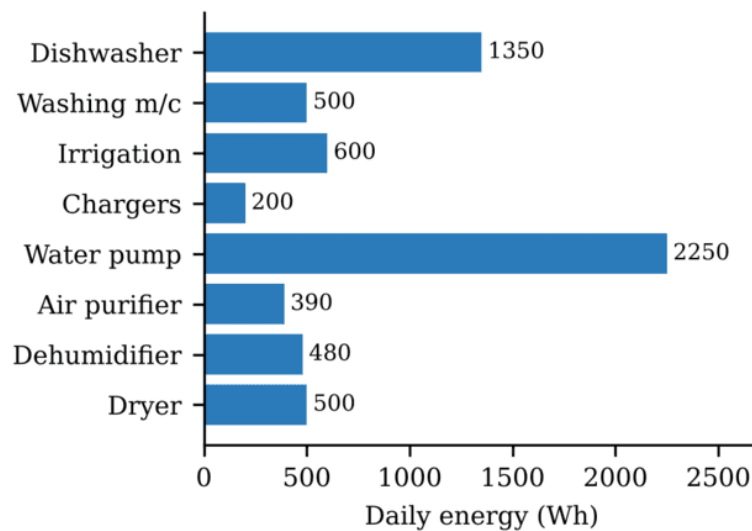
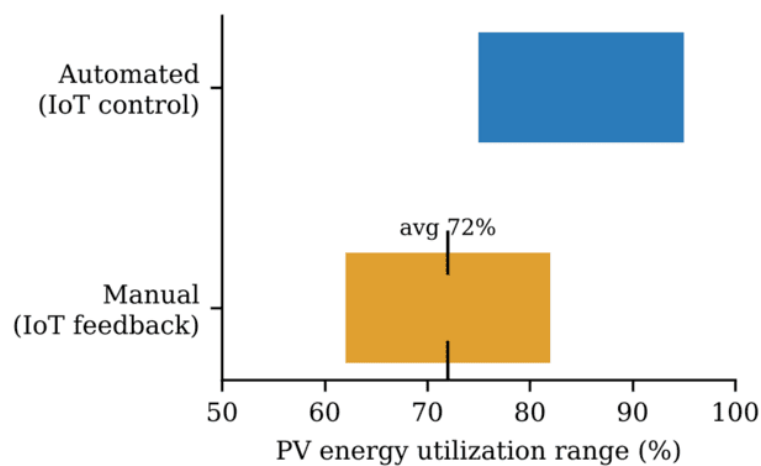
The system in [5] costs 63.04 EUR and the system in [4] about 73 EUR with a daily consumption of 13.5 Wh. The SCADA node in [3] costs CAD 94.5 with zero running cost and 3.19 W running power, and a related earlier open-source SCADA consumed 5.0 W [2]. Currencies differ, so costs are compared within their own currency only.

Table 6 - Cost and consumption of low-cost systems

Ref.	Cost	Consumption
[3]	CAD 94.5	3.19 W
[4]	about 73 EUR	13.5 Wh/day
[5]	63.04 EUR	Not stated

5.4 Load automation [6]

Five 125 W panels give a daily energy of $125 \times 5 \times 9.916 = 6197.5$ Wh, close to the planned daily load of 6270 Wh (Figure 9). Simulation gave 62-82 % utilization (average 72 %) when users react manually to IoT feedback and 75-95 % with automatic switching, which also lowered grid use (Figure 10). The result is simulated, not measured.

**Fig. 9 - Daily energy of the proposed loads (data from [6])****Fig. 10 - PV utilization with manual and automated operation (data from [6])**

Challenges and future scope

Common limits across the reviewed works are: tests on small single-string systems [3], [4], [5]; short Wi-Fi range of about 100 m [4]; no data security on the web pages [4], [5]; short test duration [3]; and manual cleaning decisions. Dust data were collected with artificial dust on one 5.5 W panel [1].

Future work includes machine-learning prediction of cleaning time using Random Forest, One-Class SVM or autoencoders [1]; image analysis of dust using camera nodes [2]; LoRa for long range [4]; cloud storage and cyber-security [3]; fault detection under partial shading [5]; and testing automation on real rooftops [6].

Conclusions

IoT makes PV monitoring cheap and accessible. The primary study [1] showed that dust seriously reduces output, with fly ash (49.16 % average loss) far worse than normal dust (17.46 %), and that real-time Blynk monitoring reveals the loss as it happens. The reference studies show that the same open-source approach scales to SCADA, fault diagnosis, MPPT monitoring and load automation at 63 to 95 currency units per node. Larger field trials, security and predictive analytics are the main gaps.

References

- [1] R. Pimpalkar, A. Sahu and A. Roy, "A smart solar PV monitoring system using internet of things (IoT)," *Concurrent Engineering: Research and Applications*, vol. 33, no. 1-4, pp. 50-60, 2025, doi:10.1177/1063293X251325155.
- [2] W. He, M. J. A. Baig and M. T. Iqbal, "An Internet of Things-Supervisory Control and Data Acquisition (IoT-SCADA) architecture for photovoltaic system monitoring, control, and inspection in real time," *Electronics*, vol. 14, no. 1, art. 42, 2025, doi:10.3390/electronics14010042.
- [3] W. He, M. J. A. Baig and M. T. Iqbal, "An open-source supervisory control and data acquisition architecture for photovoltaic system monitoring using ESP32, Banana Pi M4, and Node-RED," *Energies*, vol. 17, no. 10, art. 2295, 2024, doi:10.3390/en17102295.
- [4] A. Hamied, A. Mellit, M. Benghanem and S. Boubaker, "IoT-based low-cost photovoltaic monitoring for a greenhouse farm in an arid region," *Energies*, vol. 16, no. 9, art. 3860, 2023, doi:10.3390/en16093860.
- [5] N. Rouibah, L. Barazane, M. Benghanem and A. Mellit, "IoT-based low-cost prototype for online monitoring of maximum output power of domestic photovoltaic systems," *ETRI Journal*, vol. 43, no. 3, 2021, doi:10.4218/etrij.2019-0537.
- [6] M. Uzair, S. Al-Kafrawi, K. Al-Janadi and I. Al-Bulushi, "A low-cost, real-time rooftop IoT-based photovoltaic (PV) system for energy management and home automation," *Energy Engineering*, vol. 119, no. 1, 2022, doi:10.32604/EE.2022.016411.