energies

Article

Cloud Computing and IoT Based Intelligent Monitoring System for Photovoltaic Plants Using Machine Learning Techniques

Masoud Emamian [1], Aref Eskandari [1], Mohammadreza Aghaei [2,3,*], Amir Nedaei [1], Amirmohammad Moradi Sizkouhi [4] and Jafar Milimonfared [1]

1 Department of Electrical Engineering, Amirkabir University of Technology, Tehran 15119-43943, Iran; masoud.emami98@aut.ac.ir (M.E.); skandary.aref69@gmail.com (A.E.); amirnedaei@aut.ac.ir (A.N.); monfared@aut.ac.ir (J.M.)
2 Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology (NTNU), 6009 Ålesund, Norway
3 Department of Sustainable Systems Engineering (INATECH), University of Freiburg, 79110 Freiburg, Germany
4 Department of Electrical and Computer Engineering, Concordia University, Montréal, QC H3G 1M8, Canada; amir.94.eng@gmail.com
* Correspondence: mohammadreza.aghaei@ntnu.no

Abstract: This paper proposes an Intelligent Monitoring System (IMS) for Photovoltaic (PV) systems using affordable and cost-efficient hardware and also lightweight software that is capable of being easily implemented in different locations and having the capability to be installed in different types of PV power plants. IMS uses the Internet of Things (IoT) platform for handling data as well as Interoperability and Communication among the devices and components in the IMS. Moreover, IMS includes a personal cloud server for computing and storing the acquired data of PV systems. The IMS also consists of a web monitor system via some open-source and lightweight software that displays the information to multiple users. The IMS uses deep ensemble models for fault detection and power prediction in PV systems. A remarkable ability of the IMS is the prediction of the output power of the PV system to increase energy yield and identify malfunctions in PV plants. To this end, a long short-term memory (LSTM) ensemble neural network is developed to predict the output power of PV systems under different environmental conditions. On the other hand, the IMS uses machine learning-based models to detect numerous faults in PV systems. The fault diagnostic of IMS is based on the following stages. Firstly, major features are elicited through an analysis of Current–Voltage (I–V) characteristic curve under different faulty and normal events. Second, an ensemble learning model including Naive Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) is used for detecting and classifying fault events. To enhance the performance in the process of fault detection, a feature selection algorithm is also applied. A PV system has been designed and implemented for testing and validating the IMS under real conditions. IMS is an interoperable, scalable, and replicable solution for holistic monitoring of PV plant from data acquisition, storing, pre-and post-processing to malfunction and failure diagnosis, performance and energy yield assessment, and output power prediction.

Keywords: cloud computing; ensemble learning; intelligent monitoring system; internet of things; power prediction; fault detection; autonomous monitoring

check for updates

Citation: Emamian, M.; Eskandari, A.; Aghaei, M.; Nedaei, A.; Sizkouhi, A.M.; Milimonfared, J. Cloud Computing and IoT Based Intelligent Monitoring System for Photovoltaic Plants Using Machine Learning Techniques. *Energies* **2022**, *15*, 3014. https://doi.org/10.3390/en15093014

Academic Editor: Abu-Siada Ahmed

Received: 23 February 2022
Accepted: 18 April 2022
Published: 20 April 2022

1. Introduction

According to Snapshot of Global PV Markets 2021, the total cumulative photovoltaics (PV) installations escalated significantly to a peak of 758.9 GW in 2020 and will presumably exceed 1 TW in 2022, considering a medium scenario where cases such as the COVID-19 pandemic are also taken into account [1]. With this growing trend in PV installations, the systems' stability, reliability, and safe operation are of vital importance. The quality and commercial attractiveness of a PV system is primarily determined by its performance in the field, cost, and lifetime, to each of which the PV module significantly contributes [2,3]. However, PV systems experience various faults due to environmental conditions, human errors, and equipment failure during their service life [4,5].

PV monitoring systems have received remarkable attention over the last two decades due to the inability to diagnose operation and maintenance problems in the absence of adequate monitoring that have led to a dramatic shortening of the useful life of PV systems [6,7]. To build an effective PV monitoring system and increase the amount of produced power to an acceptable level, one should understand the challenges of monitoring a PV system. First, the required commercial data acquisition existing on the market constitutes a large part of the cost of a monitoring system. Second, the common approaches used in monitoring a PV plant require a personal computer (PC) as a computation system to run special software in real-time. Moreover, the output power of a PV system may vary due to the influence of various failures and defects on PV modules. As a result, a human operator should immediately identify and repair any detected failure on any PV panel, regardless of which system they operate on. Hence, to address the aforementioned challenges, a smart monitoring system and an accurate prediction model are required [8].

Accordingly, in recent years, many studies related to the PV monitoring systems have been carried out [9]. In this regard, N. Forero et al. [10] presented a monitoring system using a commercial data logger to save and transfer electrical and environmental data to LabVIEW (Laboratory Virtual Instrument Engineering Workbench). The PV monitoring system developed in [11,12] uses a wireless data acquisition, which is a system consisting of a PIC microcontroller and LabVIEW packages for the user interface. However, the proposed PIC microcontroller algorithm uses complicated RISC (Reduced Instruction Set Computer) such that the code is not user friendly and consequently, the LabVIEW software is not developed as an open-source software package. A. Chouder et al. [11] developed a PV monitoring system in Algeria using LabVIEW, in which data are transferred between PC and LabVIEW. The system employs an Agilent 34970A 16-channel data logger to process and transfer climatic and electrical data to a server for preparing an Excel report. As a considerable disadvantage, the monitoring system utilizes a PC as the main server; therefore, the flexibility of the system is reduced dramatically. In [12], an ET-7017 device is applied as data acquisition system, which has 20 analog single-ended channels. Moreover, the TCP/IP protocol is applied for transferring data, and then shows the PV plant information in a web application. Thus, the PV parameter monitoring system, which has been developed by many researchers, is still relatively complicated and requires a high cost of production. This is largely due to the utilization of expensive pyranometers in the entry of the photovoltaic system to measure irradiance, the use of expensive controllers, and the use of heavy weight software such as LABVIEW [13,14]. J. Han et al. [15] implemented a PV monitoring system based on Power Line Communication (PLC) for communication. In [16], a PV module monitoring system is designed based on PLC to detect fault at the panel-level. However, the PLC modules used in PV panels are expensive and not affordable for small PV plants. In another research study [17], the performance of a PV monitoring system was investigated in Doha. In this project, the sensors collect meteorological and electrical data using the data logger via a PCB board. Later on, data is transmitted through a ZigBee device to the PC for processing. The limitation of the monitoring systems is to employ expensive equipment.

However, recent years have witnessed a huge growth in cost optimization of high demand hardware used for PV monitoring. Pereira et al. [18] have designed a monitoring system based on the RaspberryPi platform, and a cloud service for a decentralized PV plant. The data logger includes a PIC microcontroller to convert A/D (Analog-to-digital) converter and the RaspberryPi board. The advantage of proposed system was the use of lightweight software. However, it suffers from hardware complexity. An Internet of Things (IoT) platform for a monitoring system in solar systems with 3G connectivity was developed in [19,20]. The collected information is sent to a server by 3G connectivity to be stored on a cloud storage space and is then represented in a web application. Closeness to a 3G station is the critical limiting factor that must be considered in this approach.

Besides, over the recent years, researchers have implemented and designed different machine learning-based fault detection and output power prediction models for PV plants. Power output prediction of PV plants plays a crucial role in evaluating the performance of a PV system [21]. It should be noted that the output power in PV systems depends on climatic factors such as radiation, humidity, and temperature, which can lead to instability in the PV plants power output. There is a wide range of methods available for PV output power prediction. For instance, Samara et al. [22] have introduced a monitoring system for PV panels using the artificial neural network (ANN) approach in order to predict the output power for diagnosing degradation. The ANN model is not applicable if the data is noisy. In [23], a comparative study on SVM and KNN is conducted based on weather classification models for short-term PV power forecasting. A major drawback of the proposed method is using a KNN model, because it does not perform properly with a large and high dimensional dataset. Raza et al. [24] proposed an ensemble framework for the day-ahead forecasting of PV output power in smart grids. This model demonstrates that the proposed framework improves the accuracy in comparison with individual and benchmark models. This model has limitations for time series data [25]. Furthermore, in recent years, studies have been carried out on PV systems fault diagnosis using ML techniques [26]. B. Basnet et al. [27] proposed the multilayer perceptron (MLP), followed by a supervised learning approach. However, this is restricted to specific environmental conditions, since the study focuses on collecting data only in the winter. Besides, Ref. [28] suggests a fault detection algorithm, which has been based on theoretical curves modelling and fuzzy classification system using LabVIEW. Generally, the approaches seem to be computationally demanding and relatively expensive. In Ref. [29], a method based on the assessment of three coefficients to detect and classify line-line and open-circuit faults is proposed, in which real measured and simulation model predicted coefficients are compared. However, despite simplicity, the model seems to be vulnerable and susceptible to an ill-suited threshold setting [30]. Besides, there are some other studies which suffer from the ignorance of the shift-invariant properties of the vibration data and, therefore, in feature extraction the focus of attention is mostly on the frequency spectrum data rather than the raw vibration data [31]. In fact, there are many studies in which different sorts of faults are detected based on a comparison between the measured and modeled outputs and, afterwards, the diagnosis is performed [32], which can be over- or underestimated unless the model is perfectly accurate or, otherwise, it can cause fault detection in turn.

In this study, we propose the Intelligent Monitoring System (IMS), consisting of an IoT platform, a cloud service, and a web monitoring software for end-users. Accordingly, we present a cloud service for data processing (cloud computing), cloud storage for recording data, and use an IoT platform for data transmission. Moreover, the proposed method employs a long short-term memory (LSTM) ensemble neural network to predict the output power PV systems for one day ahead. The LSTM model is appropriate for sequential and time series problems. The proposed PV monitoring system forecasts output power of PV systems, analyzes information, and creates detailed reports automatically for repair and maintenance in PV plants. Moreover, another ensemble learning-based model is utilized for fault diagnosis of PV arrays to maximize the classification accuracy and the diagnostic performance as well as to attenuate and lighten the computational burden [33]. In an early study [33], we proposed a novel method to detect line-line faults. However, it should be noted that in current study, this model is based on the IoT platform and cloud computing. Moreover, this model is designed to detect and classify open circuit faults, array degradation faults, and string degradation faults using an ensemble learning model that contains three single algorithms, namely Naive Bayes (NB), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). It is worth mentioning that, in general, we have selected these three algorithms since they are firstly proved to be the most powerful methods for detecting and classifying various faults in PV systems and, secondly, they are perfectly capable of dealing with multi-class and binary classification problems. To be precise, Support Vector Machine (SVM) is selected to solve complex classification problems using kernel tricks. Besides, we have deployed Naive Bayes (NB), which performs very effectively with a small amount of data, especially in the process of training. Finally, K-Nearest Neighbors (KNN) is nominated because it has low bias, but that also means it can have high variance. Thus, the main objective is the development of an IMS to reduce hardware equipment for which the proposed method uses open-source and lightweight software and also cost-efficient hardware. This system is appropriate for a stand-alone system and remote regions and has the flexibility for implementation on different types of PV plants. The scope of the IMS system is to detect and classify three electrical faults and predict the output power in PV arrays.

The rest of this paper is organized as follows. Section 2 describes the IMS architecture and discusses the details of hardware and software development. The architecture of the developed ensemble learning techniques is described in Section 3. Section 4 focuses on the simulations and experiment results to validate the performance of the proposed method. The paper is concluded in Section 5.

2. The Proposed Intelligent Monitoring System (IMS)

The aim of this study is to design an affordable and high precision monitoring system in order to perform the precise monitoring of PV plants. Consequently, an IMS has been proposed and designed to predict the output power of PV modules on the day ahead and diagnose fault events for improving the performance of the PV plant.

Figure 1 depicts the structure of the IMS proposed in this paper as well as the location of environmental and electrical sensors. The IMS collects meteorological and electrical data through sensors installed on PV modules, PV arrays, and the environment. As shown in Figure 1, the data logger consists of ESP8266 and Microcontroller boards. The electrical data is sent to the ESP8266 module for initial processing. This step is required for data preparation to be transferred to the server. The ESP8266 is a cost-efficient mini-Wi-Fi device based on ESP8266EX, which is popular for IoT applications. The ESP8266 module is able to be connected to the internet via the access point to post data to the cloud service for recording and ultimate processing (cloud computing). The device is set up in order to connect different types of internet access points.

Having quick access to the internet and accordingly the web monitoring system, which can provide analysis, information, predictions, and numerous reports, the monitoring service has noticeably been facilitated for different people such as owners, operators, end-users, etc. through gadgets and smart devices such as computers, mobile phones, etc. Moreover, it is worthwhile be noted that the proposed IMS takes advantage of open-source programming languages for developing different parts of the architecture. The ESP8266 module is developed via C++ programming language to collect and transfer data to the cloud computing server. On the server side, the flask framework is used as the backend system. Backend development is defined as the server side of an application. For front-end of the monitoring system, we have used HTML, CSS, bootstrap, and JavaScript to develop and design the web monitoring systems. Front-end is related to anything users interact with on the application or web browser. Moreover, different Python packages (Flask, Keras, TensorFlow, etc.) have been used to implement the ensemble learning algorithms.

The configuration of the developed IMS includes five crucial components, as follows:

- IMS structure;
- IoT platform;
- Cloud data logger;
- Cloud server;
- Web monitor.

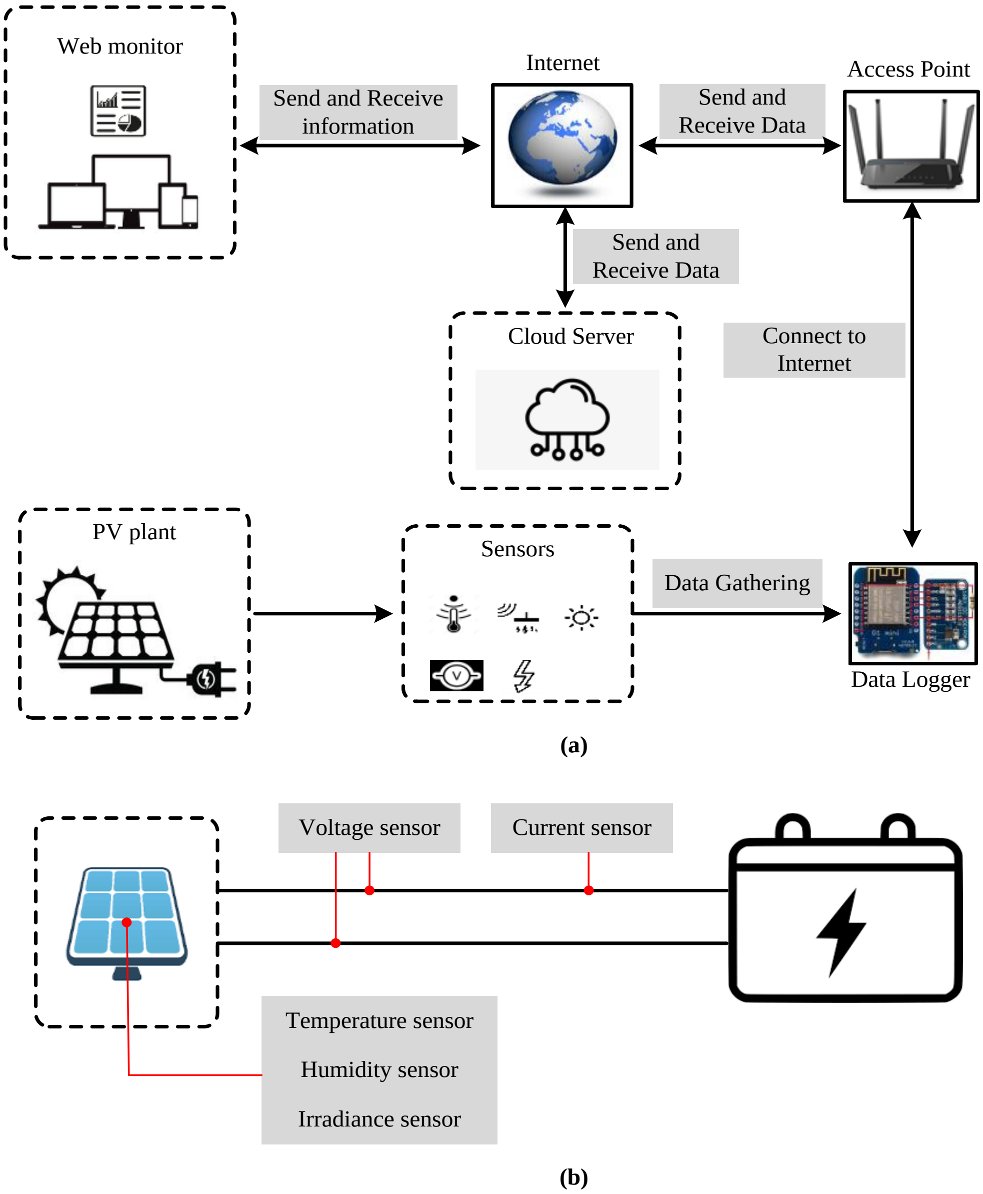


Figure 1. (**a**) An overview of Intelligent Monitoring System (ISM) architecture; (**b**) sensors distribution.

Given that the purpose of this study is to transmit information via the common communication protocol and to use the data to verify the functionality of the PV plant, we hence did not focus on the issue of security and delay or conflict of information. However, to prevent disruption, we used a backup system to collect daily data, and data will be replaced in the event of disruption after the system is fixed. It should be noted that we have highly considered the data privacy of the end-users in IMS by keeping the data at the PV plant level. The data measurement can be continued or periodic upon the request of the plant's owners.

2.1. IMS Architecture

Figure 2 provides an in-depth overview of how the proposed monitoring system has been arranged. In this experimental research, a stand-alone PV system has been deployed to test and analyze the performance of the IMS. As shown in Figure 2, the designed IMS consists of PV modules, microcontrollers, a data logger, and sensors with wired connections. The equipment used in hardware section include 18 PV modules with 10 W total output power, a resistance load, an access point, mini-Wi-Fi device based on

ESP8266EX, a microcontroller, and different sensors, which have been listed in Table 1. By using the presented IMS structure, different PV monitoring tasks, e.g., data collection, can be done quickly and efficiently through the connected microcontroller to a computational system via the internet.

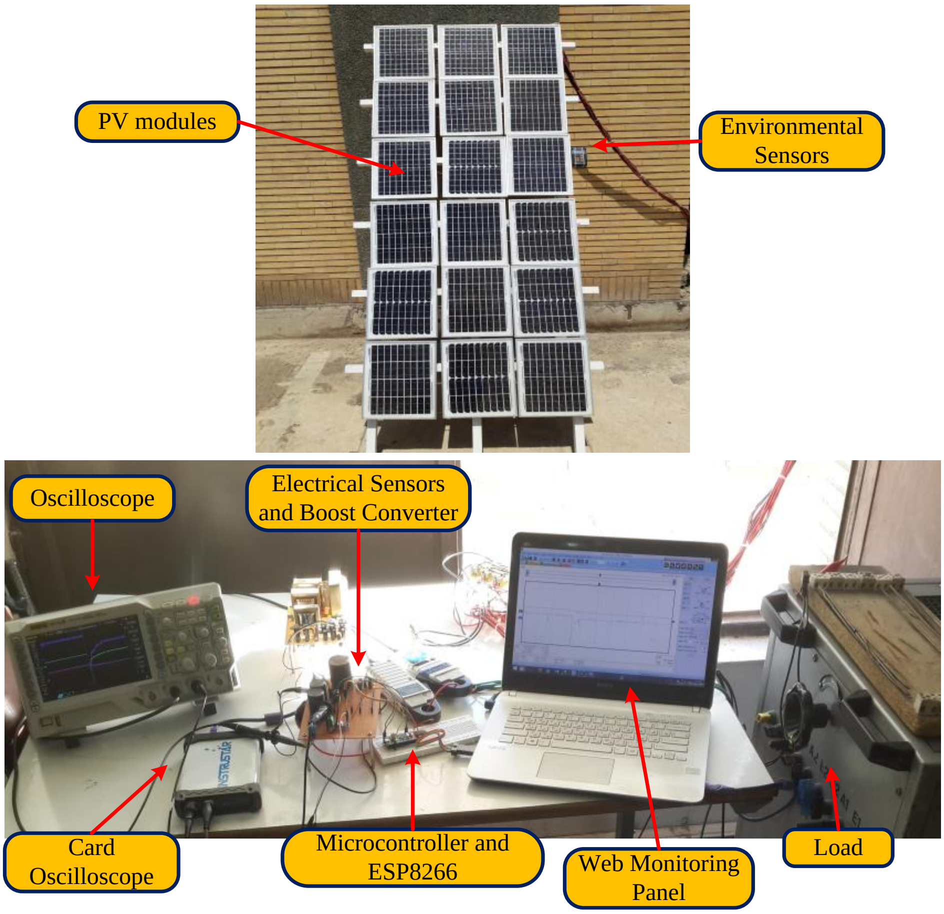


Figure 2. PV experimental setup.

Table 1. Characteristics of the applied sensors.

Module Name	Measurement Type	Measurement Range	Accuracy
DHT22	Humidity	0–100%	2–5%
	Temperature	−40 to 80 °C	±0.5 °C
ACS712	Current	0–5 volt	1%
Voltage divider	Voltage	-	-

2.2. Internet of Thing (IoT) Platform

The IoT is a new technology that is envisioned as a global network of interconnected machines, devices, objects, people, and even animals interacting with each other and transferring and sharing data over the network. IoT technology is one of the most important technologies in the future and has attracted the attention of a wide range of industries [34]. A flowchart demonstrating the process of the IoT platform process in the proposed IMS is provided in Figure 3. In the first step, the IoT file is loaded by the microcontroller. The IoT file includes posts and gets commands for data transfer between the cloud and the microcontroller. After uploading the file to the microcontroller, using an Internet connection, the system then connects to the server and executes the script (.js file) on the server. In case the connection is lost, the platform starts transferring data from the microcontroller to the server again for recording and processing. If the data transfer fails, an URL error will be displayed to the user.

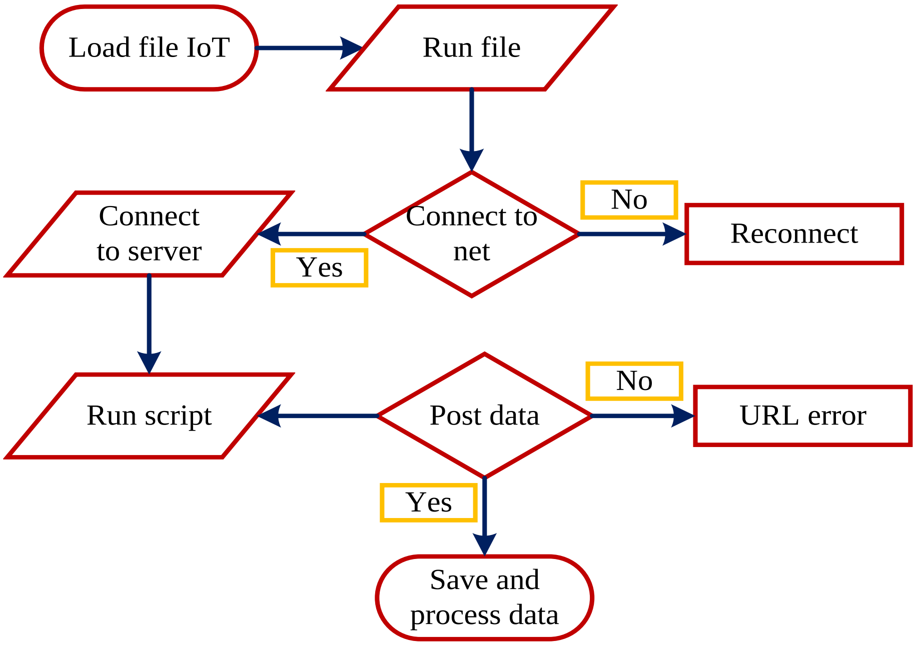


Figure 3. The flowchart of data transfer by the IoT platform.

2.3. Cloud Server

This paper focuses on powerful cloud tools, which are presented by cloud servers such as coding packages, cybersecurity, admin page, process units, etc. Figure 4 illustrates the services provided by a cloud server. Service models consist of three types, namely Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [35]. In this study, the SaaS model (Google Cloud) has been used for the cloud computation unit. The user has access to control the deployment of end applications. However, the user is not allowed to control the operating infrastructures. Cloud computing is a model for enabling ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly released with minimal management effort or service provider interaction.

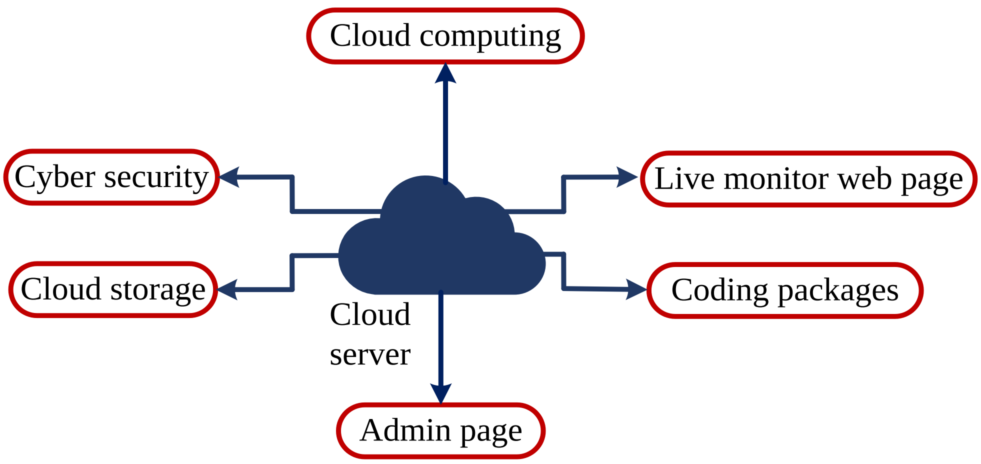


Figure 4. An overview of the cloud service's components.

2.4. Cloud Data Logger

Cloud data loggers communicate over the Internet connection with a remote cloud-based server, which can be deployed anywhere in the world. The cloud-based server uses the cloud data logger as a container for recording data. Users can take advantage of any devices supporting a web browser to access data on the cloud. This paper proposes a new structure for the data logger based on an open-source and lightweight software and a customized cost-efficient hardware. Figure 5 shows the steps of the data logger used in the IMS as a flowchart. In the first step, the data logger loads the file in .c format and later on, the file is executed on the microcontroller. The next step is to connect to Google Script. If there is any problem with the connection, the system tries to reconnect to the server. Later on, the script is executed by Google Cloud, so that the ID related to the storage file is checked with Google ID. If the IDs are the same, the microcontroller is connected to the cloud, otherwise, the URL error will be displayed. Finally, the system checks the authenticity of the ID. Once the ID is correct, it starts receiving data from the microcontroller. The data is recorded as a CSV file and displayed on the Google Sheet simultaneously.

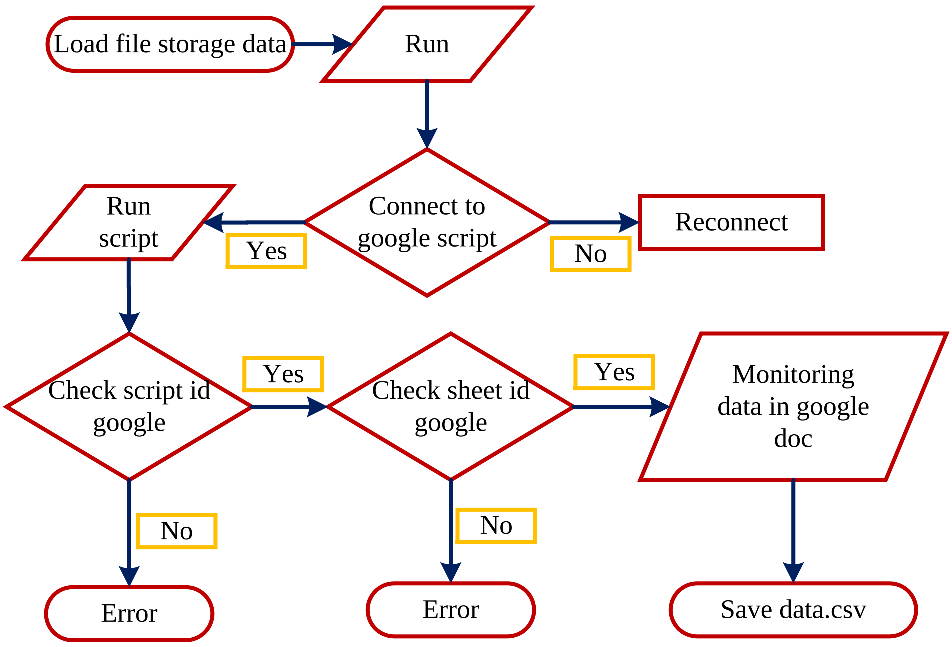


Figure 5. Data storage by cloud service flowchart.

2.5. Web Monitor

The term 'Web monitoring' signifies the process in which the information that end-users can interact with on a website or a web application via the internet is systematically analyzed and displayed (See Figure 6 for a general overview of a web monitoring process). Accordingly, the cloud server loads the scripts, which is followed by receiving a CSV-format version of the data file from the database and subsequently posting it to the backend for further processing. The backend is responsible for running programs in scripting languages like Python. The task includes execution of the ensemble learning algorithms code on the flask, users' identification, and data processing in the backend. Finally, the processed data is displayed in the frontend. The task of the frontend layer is to run web languages such as HTML, CSS, and JavaScript in web browsers. Figure 7 shows the overview of the frontend layer used for data display to users. If the error occurs in the process layers, the user receives the URL error through the browser.

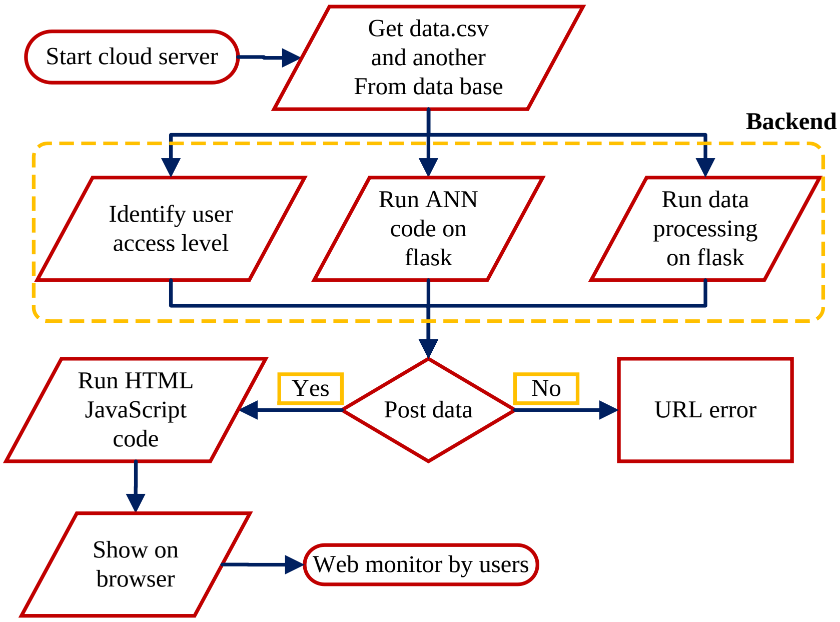


Figure 6. Performance intelligent web monitor flowchart.

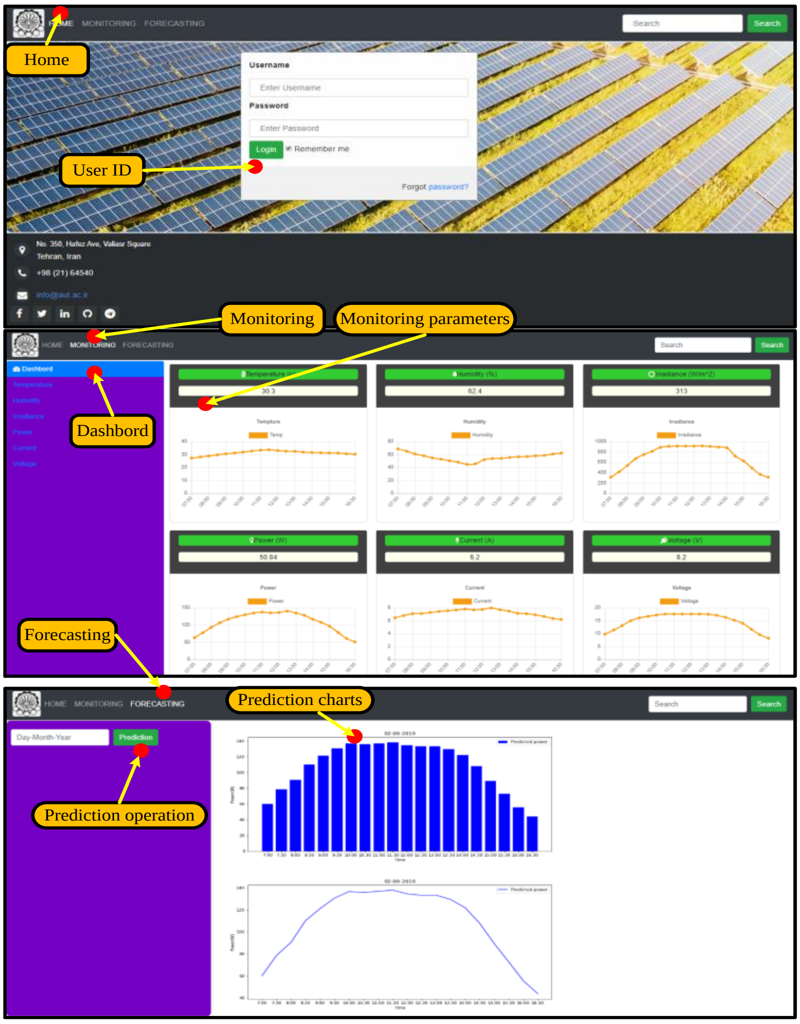


Figure 7. Display panel Intelligent Monitoring System (IMS).

3. The IMS Using Machine Learning Techniques

To develop the IMS in order to obtain an accurate prediction of the output power and a precise detection of PV faults, we have applied novel machine learning techniques (MLTs) in a way that a long short-term memory (LSTM) model helps the IMS in the process of power prediction while, on the other side, an ensemble learning (EN) classifier which is based on the I-V curve measurement of the PV arrays is employed to provide a more accurate detection of the faults.

3.1. Power Forecasting Architecture

Recurrent neural networks (RNNs) are a popular deep learning model that is widely used for the prediction of time series and sequential instances. A RNN model includes a recursive loop, so that the model stores the past information to update and enhance the model for future prediction. Besides the efficiency of RNN in long-term forecasting, it suffers from two major drawbacks including vanishing and exploding gradients. In RNNs, vanishing and exploding gradients can result in an unstable network that is not able to learn features sufficiently from training data. Consequently, the network cannot learn over long input sequences of data. To overcome these drawbacks, LSTM is introduced, which is an improved version of RNNs.

This study aims to achieve an efficient prediction model by composing ensemble neural networks and LSTM. The ensemble methods improve the dynamics and enhance the accuracy of the neural network models. Accordingly, this experimental research has developed an LSTM ensemble architecture for output power prediction of PV Plants. In the following, we describe three main components of the LSTM model, which are the structure of LSTM, LSTM ensemble architecture, and the evaluation methods.

3.1.1. LSTM Structure

The LSTM models are a modified version of RNNs, which resolve the vanishing and exploding gradient problems. These models recall past data more easily than traditional RNNs and are suitable for predictive time series instances [36].

Figure 8 shows the LSTM architecture, which includes four layers in its hidden layers. RNN models are a specific type of LSTM models with a major difference, being the existence of gates in RNNs. The LSTM neural networks consist of three types of gates, namely, the input, output, and forget gates. The following equations represent the methodology of layers, operators, and gates of the LSTM model, as follows:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{1}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{2}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + W_{cm}m_{t-1} + b_c), \tag{3}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_c), \tag{4}$$

$$h_t = o_t \odot \phi(c_t). \tag{5}$$

In Equations (1)–(5), x_t represents the current input, c_{t-1} is the memory from the last LSTM unit, h_{t-1} denotes the output of the last LSTM unit, c_t shows the new updated memory, and h_t signifies the current output. Also, according to Equations (1)–(5), W_{mn} denotes the connection weight from gate m to n. The parameter b is a bias parameter for the training model, where $n \in \{x, h\}$, and $m \in \{i, f, o, c\}$. Moreover, the operator $\odot$ indicates the element-wise product (Hadamard product), σ indicates the logistic sigmoid function, and Φ indicates the network's output activation function. These functions are calculated as follows:

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{6}$$

$$\Phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{7}$$

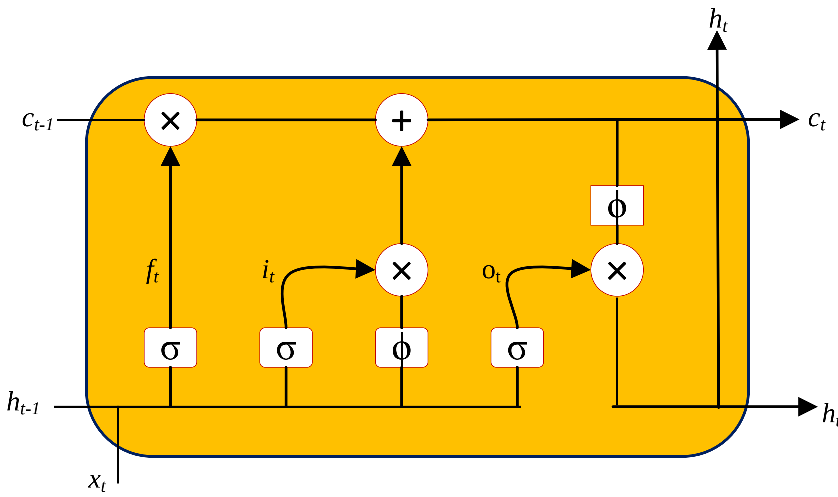


Figure 8. LSTM neural network structure.

3.1.2. LSTM Ensemble Architecture

The ensemble methods have been introduced for improving the performance and learning accuracy of the machine learning models. This paper uses an enhanced model of the ensemble LSTM for the output power prediction of PV plants [37]. The architecture of the ensemble LSTM is developed based on some single NNs used as predictor models. Figure 9 shows the three layers of the ensemble model including input layer, training layer, and forecasting layer (output layer). In the input layer, the electrical and environmental data are given to the LSTM network. Next, inside the training layer, each LSTM model is trained separately. Finally, for final output prediction, a particular weight coefficient is assigned to each model's output for output forecasting of the ensemble LSTM calculation. In this model, the weight of the prediction and output forecasting are according to Equation (8), which considers the entire M LSTM models that are presented in a set of models. The ensemble prediction results for the time series are represented as $\left(y^1\ ,\ y^2\ \ldots,\ y^N\right)$ with N observations. Parameter $\hat{y}_j^i$ signifies the prediction output obtained using of jth LSTM model (at the ith time interval) and w_j is subject to weights composition. In Equation (8), each weight w_j is related to a prediction output of the LSTM model, where, $0\ \leq\ w_j\ \leq\ 1$ and $\sum_{j=1}^{n} w_j = 1$.

$$\hat{y}_i = \sum_{j=1}^{n} w_j \hat{y}_j^i \text{ for } i = 1, \ldots, N. \quad (8)$$

Weight combination is one of the most significant parameters in the efficiency prediction of our LSTM ensemble model. According to Equation (9), w_j of weight compounds $(j = 1, \ldots, n)$ that n is a member of the basic LSTM ensemble network. Based on the experiments performed on the LSTM ensemble network, M = 5 has been selected as the best choice. Equation (9) is used to calculate the weights of the power output prediction of the basic LSTM model for the day $d+1$. Accordingly, e_i^{norm} is the normalized error of single ith LSTM model over the number of days, and j is number of LSTM ensemble network models. The e parameter is considered as mean absolute error (MAE) in the first LSTM ensemble network, and as root mean square error (RMSE) error in the second LSTM ensemble network. The first LSTM ensemble network is represented by the symbol EN_1 and the second by the symbol EN_2, and they are computed as follows:

$$W_i^{d+1} = \frac{1 - e_i^{norm}}{\sum_{j=1}^{M} 1 - e_j^{norm}}. \quad (9)$$

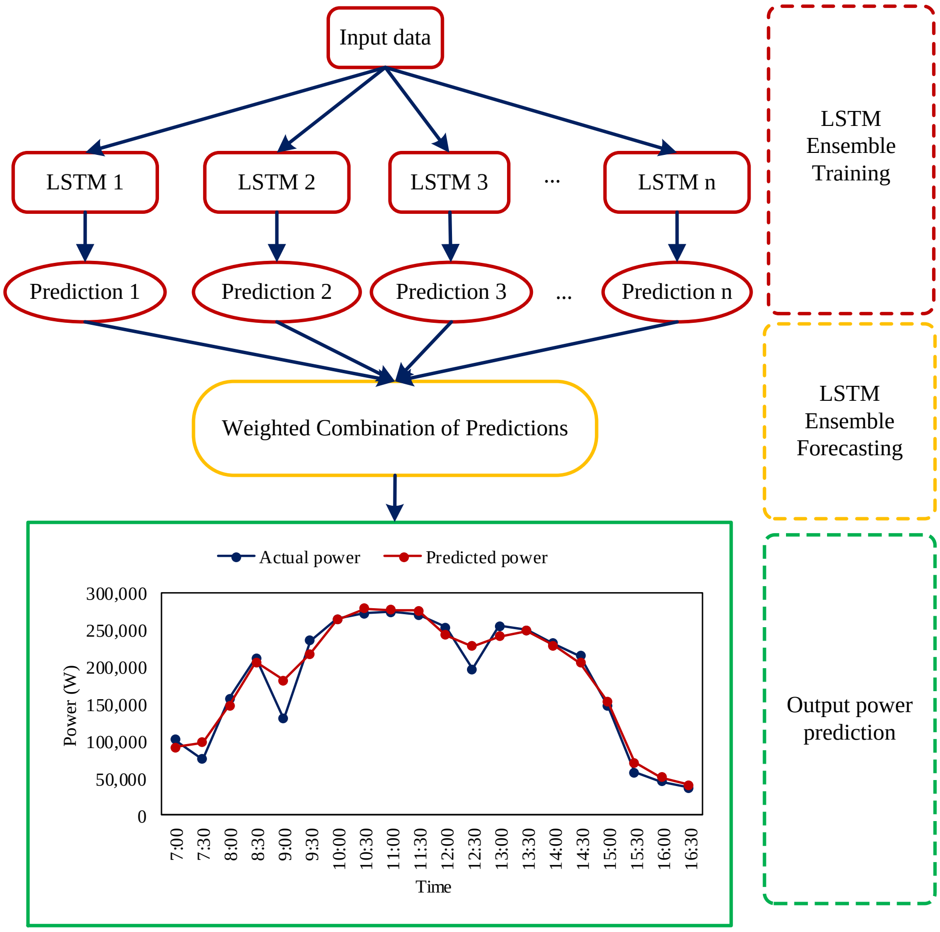


Figure 9. LSTM ensemble architecture.

3.1.3. Evaluation Methods

In this experimental study, to evaluate the performance of the regression models, three criteria, namely RMSE, MAE, and Mean Absolute Percentage Error (MAPE) have been used. In Equations (10)–(12), P_i and $\hat{P}_i$ are the real output power and predicted output power for the ith day in PV system. Parameter N is the number of days in testing dataset, and n is the number of time step of output power prediction for one day $(n = 20)$.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(P_i - \hat{P}_i\right)^2}{N \times n}}, \tag{10}$$

$$\text{MAE} = \frac{1}{N \times n} \sum_{i=1}^{N} \left| P_i - \hat{P}_i \right|, \tag{11}$$

$$\text{MAPE} = \frac{1}{N \times n} \sum_{i=1}^{N} \frac{\left| P_i - \hat{P}_i \right|}{P_i}. \tag{12}$$

3.2. Fault Detection and Classification

In this paper, an ensemble learning (EN) model for fault diagnosis of PV arrays is investigated in order to achieve superiority in classification accuracy. Figure 10a shows an overview of the required steps to build an EN-based fault classification model. As a supervised machine learning technique, the EN-based model relies on a certain number of fault features extracted from the raw data obtained. To achieve a high diagnostic performance with a low computational burden, a feature selection algorithm is utilized.

To evaluate the performance of the model, the labeled samples are divided into training and validation sub-sets. The training set is used to train the model, while the validation set is used to measure the performance of the model. Then, we have tested the learning algorithm by an unseen dataset with dissimilar input parameters to the original dataset.

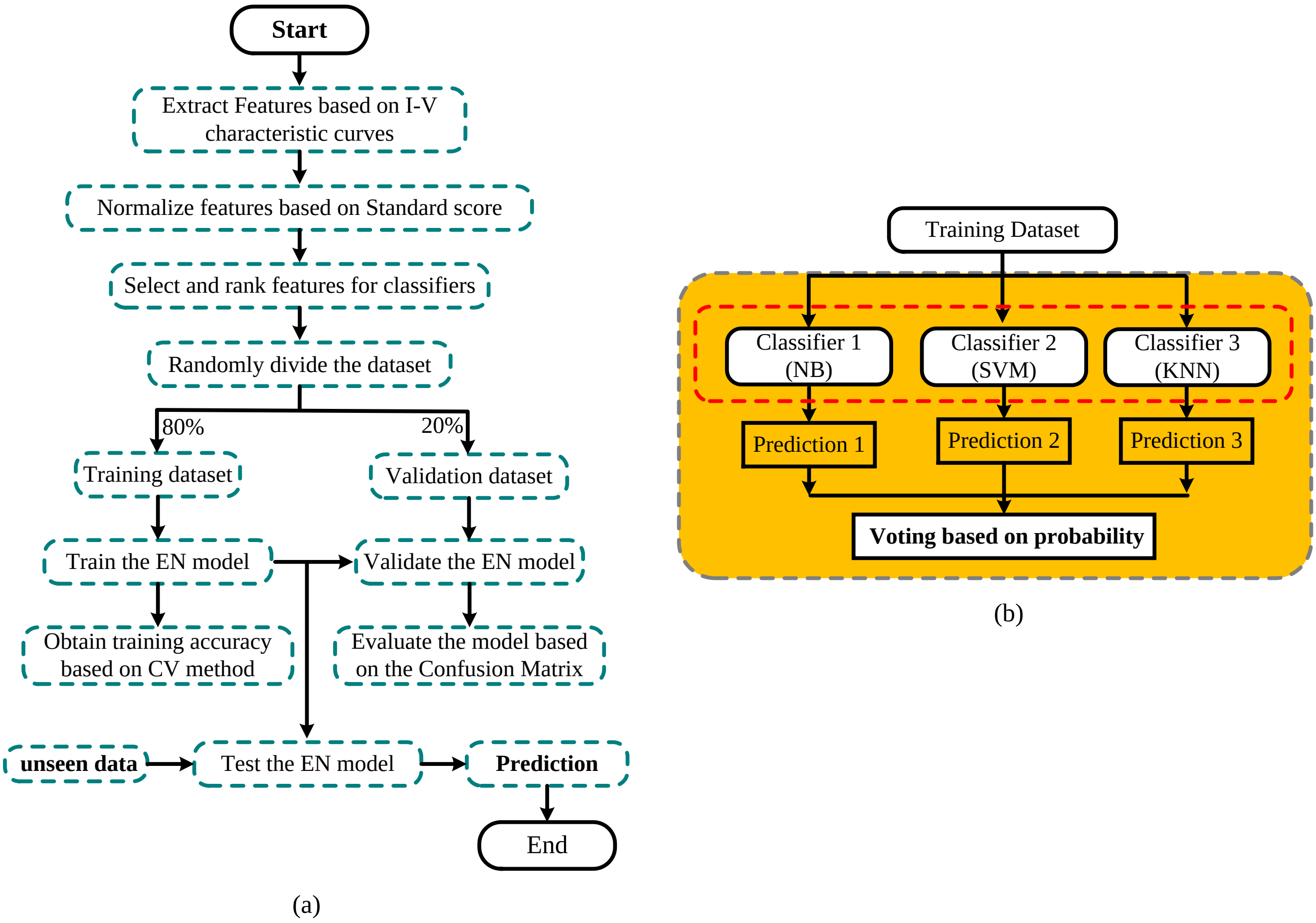


Figure 10. (**a**) The algorithm structure of the proposed method, and (**b**) flowchart of the EN model.

3.2.1. Feature Extraction

In this section, the dataset must go through a two-stage process which includes (1) faults attributes identification and (2) feature extraction. Normally, the first stage is fulfilled by a detailed analysis of the current-voltage (I–V) characteristic curve.

The output voltage and current of the PV array are recorded to capture the I-V curves. Then, in the second stage, based on a careful analysis of five specific points on the current-voltage curve, namely short circuit current, open- circuit voltage, Maximum Power Point (MPP), half short-circuit current, and half open-circuit voltage, for a more accurate fault detection, the algorithm has extracted 16 features which are listed in Table 2, normalized to standard testing conditions (STC).

As shown in Figure 11, two layers of the EN model are designed to detect any fault and classify its type. The first layer aims to detect the fault condition through a binary classification. The second layer includes the multiple classes, which classify the type of defects by a multi-label classification algorithm. There are three kinds of outputs, namely open-circuit, string, and array degradation faults.

Table 2. Extracted features from current-voltage curves.

Extracted Features		
$f_1 = I_{SC}/I_{SC(STC)}$	$f_2 = V_{OC}/V_{OC(STC)}$	$f_3 = V_{MPP}/V_{MPP(STC)}$
$f_4 = I_{MPP}/I_{MPP(STC)}$	$f_5 = I_{\frac{V_{OC}}{2}}/I_{SC(STC)}$	$f_6 = V_{\frac{I_{SC}}{2}}/V_{OC(STC)}$
$f_7 = f_4/f_3$	$f_8 = f_3/f_2$	$f_9 = f_4/f_1$
$f_{10} = (I_{MPP} - I_{SC})/(V_{MPP})$		$f_{11} = (-I_{MPP})/(V_{OC} - V_{MPP})$
$f_{12} = (I_{\frac{V_{OC}}{2}} - I_{SC})/(\frac{V_{OC}}{2})$		$f_{13} = (I_{MPP} - I_{\frac{V_{OC}}{2}})/(V_{MPP} - \frac{V_{OC}}{2})$
$f_{14} = (\frac{I_{SC}}{2} - I_{MPP})/(V_{\frac{I_{SC}}{2}} - V_{MPP})$		$f_{15} = (-\frac{I_{SC}}{2})/(V_{OC} - V_{\frac{I_{SC}}{2}})$
$f_{16} = \frac{FF}{FF_{(STC)}} \rightarrow FF = \frac{V_{MPP} \times I_{MPP}}{V_{OC} \times I_{SC}}$, $FF_{(STC)} = \frac{V_{MPP(STC)} \times I_{MPP(STC)}}{V_{OC(STC)} \times I_{SC(STC)}}$		

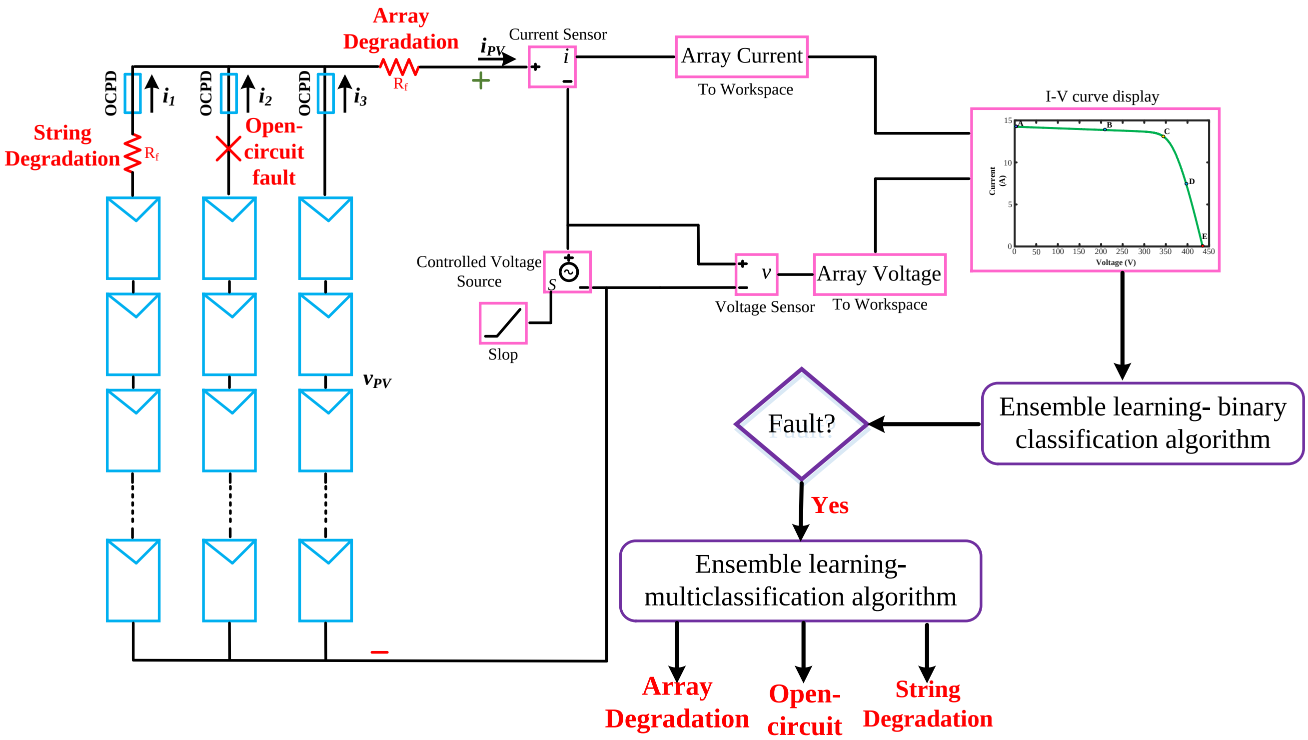


Figure 11. Overall view of the proposed EN model for fault diagnosis.

The fault detection process is as follows: the first layer identifies whether new sample data is affected by any kind of fault or not. If the sample data is defected, then the second layer is responsible for recognizing the type of the fault.

3.2.2. Feature Selection Algorithm

Feature selection is a process by which a proper subset of the original extracted features is carefully chosen by the elimination of the unrelated features to enhance the overall classification performance of the ML algorithm and to reduce the learning time. There exist different methods of selecting the appropriate features, among which, we have chosen a wrapper method and specifically a sequential algorithm owing to its concentration on the usefulness of the features based on the performance of the classifier.

The above-mentioned selection method is the Sequential Floating Forward Selection (SFFS) algorithm, mainly chosen for its potential capability of backward movement.

Initially, the SFFS starts with an empty subset and proceeds with adding the most accurate feature chosen by the classifier and eliminating all previously selected features at the end of each iteration. Subsequently, new subsets are evaluated to determine whether the prior nominated features were optimal or not. This loop is repeated over the number of features in the dataset in order to obtain the most optimal number of features with the highest accuracy.

3.2.3. The Proposed Ensemble Learning Algorithm

Using an appropriate learning algorithm is a matter of the utmost importance in ML-based fault classification. Many studies have deployed a single classification algorithm, which has culminated in low accuracy or poor performance. To cope with the aforementioned problems, in this study, we have proposed an ensemble learning algorithm which is comprised of three different classifiers, namely Support Vector Machine (SVM), Naive Bayes (NB), and K-Nearest Neighbors (KNN).

The probability of data belonging to a specific class has a large impact on the overall performance of the model for data classification. Hence, in this paper, a probabilistic ensemble model is devised to combine the merits of three previously mentioned classifiers (see Figure 10b). The process starts with predicting the class labels according to the average of probabilities predicted by each single algorithm. Then, the model calculates the average probability of every class label attributed to the sample data and eventually selects the label with the highest probability. The process is formulated in Equation (13):

$$\hat{y} = \arg\max_{j} \frac{\sum_{i=1}^{m} P_{ji}}{m}, \tag{13}$$

where $\hat{y}$ is the output predicted by the classifier and P_{ji} is the probability that class j is nominated for a sample data belonging to ith learning algorithm (m = 3). Note that j is two for fault detection and three for fault classification.

3.2.4. Evaluation Metrics

We have divided the whole dataset into validation data and unseen data to evaluate the efficiency of the learning algorithms. For this purpose, a confusion matrix helps show the prediction results of fault detection and classification, produced by every single algorithm (see Table 3). In Table 3, TP, known as true positive, is the state in which a sample is correctly predicted in the first class. The same is true for TN, standing for true negative, which signifies a correctly identified second class sample data. Moreover, FP, short for false positive, refers to a sample that is wrongly identified as a member of the first class. Adversely, a sample data categorized as FN (false negative), should have been labeled as the first class but is incorrectly classified as the second.

Table 3. Confusion matrix to evaluate the performance of the learning algorithms.

		Predicted	
		First Class	**Second Class**
Actual	**First class**	TP	FN
	Second class	FP	TN

Furthermore, based on the confusion matrix, we have defined the accuracy and F1-score measures (Equations (13) and (14), respectively) to assess the performance of the learning algorithms.

The classification accuracy is expressed in Equation (14):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}. \tag{14}$$

The F1-score, which represents the harmonic mean of recall and precision, shows more realistically how accurately the trained model performs when dealing with unseen data. It is calculated according to Equation (15) and will be later evaluated for the different classes (Ci, $i \in \{1, 2, 3\}$):

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \tag{15}$$

where Recall shows the percentage of correctly identified actual class events (see Equation (16)) and Precision is that of correctly labeled predicted class events (see Equation (17)).

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{16}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{17}$$

Moreover, in order to evaluate the performance of the proposed method, we have devised two scenarios, one with and one without the feature selection algorithm for both layers.

4. Experimental Verification

In this paper, an IMS is developed with several abilities such as power output prediction, fault detection, cloud storage, IoT application, and web monitoring. This section gives information on the experimental results and provides an explanation in two parts. The LSTM ensemble network results are presented in the first part, while the second part carries the results related to the performance of the fault detection.

4.1. Data Acquisition

The developed IMS collects data from a stand-alone PV system (see Figure 2). In the power forecasting part, the training data has many features, e.g., temperature, irradiance, humidity, power, voltage, and current. In the feature extraction step, the training data is passed through the developed technique to find the correlation between different features. Finally, the selected features are temperature, irradiance, and power. Therefore, input data includes 3 features and 20 time steps, where each time step includes 30 min intervals from 7 am to 5 pm. The entire dataset includes 7300 samples. It should be noted that our main criterion for selecting such a dataset is based on the environmental conditions of our region where the experiments were executed.

To confirm the validity of the proposed method in fault diagnosis part, we created a combination of faulty and normal events via the original dataset as well as unseen dataset. The faulty and normal panels are tested under a vast range of irradiance between 200 $\mathrm{W/m^2}$ and 1000 $\mathrm{W/m^2}$, and the temperature changes ranging from 0 °C to 40 °C.

It should be noted that the fault degradation in a PV string is executed by applying series resistor on the String #1. Moreover, as can be seen in Figure 11, a resistor is inserted in series into the output of the PV array to imitate the degradation fault. Therefore, the range of fault impedance is set to [2 Ω, 15 Ω] with a step of 3 Ω.

The open-circuit fault is emulated by setting the infinity number of the series-connection resistor. This paper demonstrates an open circuit scenario with one disconnected PV string in String #2, as illustrated in Figure 11.

In this experimental research, we have collected 433 cases in normal condition, 433 cases with open-circuit fault, 582 cases with PV string degradation fault, and 582 cases with PV array degradation fault. Once the training process is completed, layers are evaluated by an unseen dataset. This dataset is split under different operation conditions, which are combinations of irradiance levels, operation temperatures, and fault impedances (4, 8, and 12 Ω). Therefore, to assess the model more accurately, 60 normal, 60 open-circuit, 48 PV string degradation, and 48 PV array degradation conditions are obtained by the experimental setup.

4.2. Experimental Results

4.2.1. Power Forecasting Results

In this study, EN_1 and EN_2 models are compared with other deep NN models such as LSTM, MLP, and Conv1D from different perspectives. The EN-1 model is an LSTM ensemble network with an updated weight based on RMSE error. Accordingly, the EN-2 model is an LSTM ensemble network with an updated weight based on MAE error (see Table 4).

Table 4. Parameters of the ANN algorithm.

Model	Optimizer	Activation Function	Dropout	Batch Size	Epochs
LSTM	radam	relu	0.1	16	100

Figure 12 shows the performance of the trained models. According to Figure 12, by comparing the errors of the models in two metrics (RMSE and MAE), the ensemble model has a better performance than other models in both indices. For further analysis, we also list the forecasting errors evaluation considering multiple models as numerical results, see Table 5.

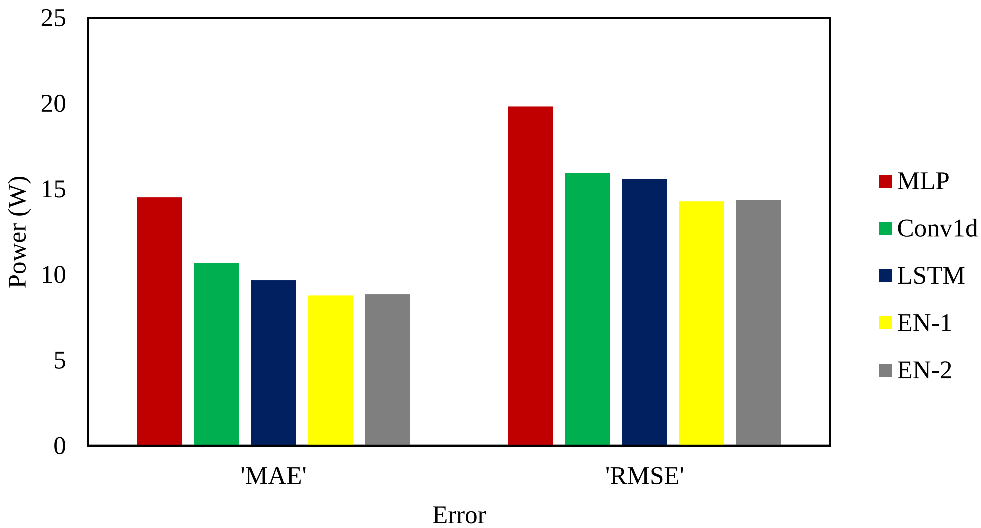


Figure 12. RMSE and MAE errors of models.

Table 5. PV output forecasting error evaluation considering multiple models.

Model	MAE(W)	MAPE (%)	RMSE(W)
MLP	14.519	4.24	19.832
Conv1d	10.69	3.47	15.941
LSTM	9.68	3.05	15.585
EN-1	8.795	2.58	14.29
EN-2	8.856	2.69	14.353

Figure 13 shows the actual power diagram of the PV plant and the predicted power by the predictor models for one day. Accordingly, the graph indicates that the output of the ensemble models is closer to the actual value compared with the rest of the models.

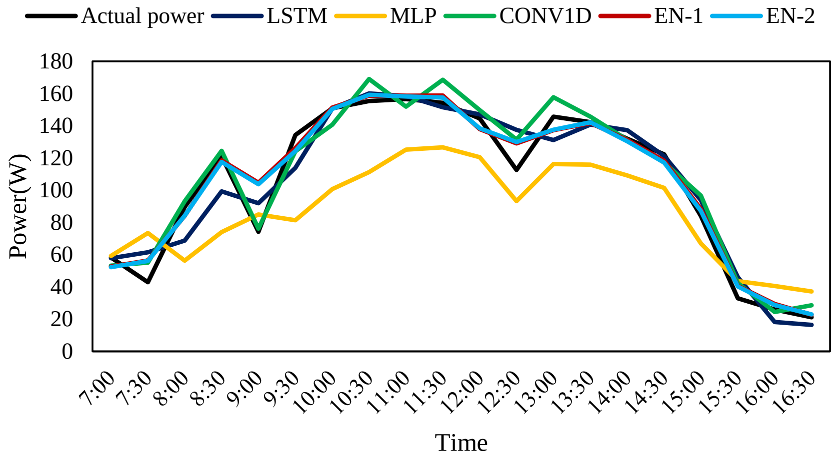
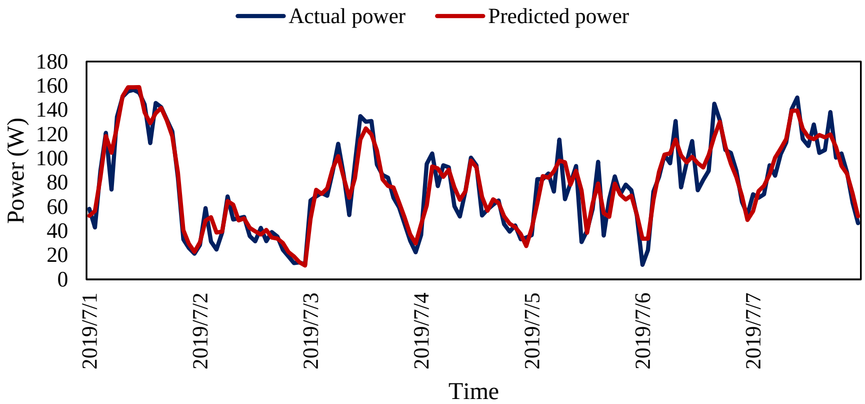
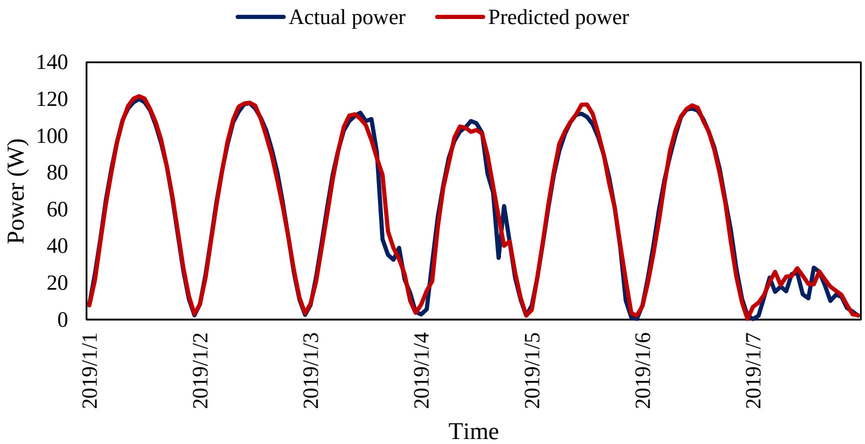


Figure 13. Forecasting and actual power diagram of deep learning models for one day.

According to the results of the output power prediction, the EN-1 model has shown better performance. Figure 14 represents the output power forecasting of the LSTM ensemble model. As shown in Figure 14a, the graphs of actual and predicted output power during one week in summer are depicted. Figure 14b illustrates the actual and predicted output power during one week in winter. As we can see, there exists a good convergence of the predicted output power to the actual output power in different seasons. Therefore, this model is able to accurately predict the output power of the PV system.



(**a**)



(**b**)

Figure 14. Real and predicted power output diagram. (**a**) The EN-1 model for one week in the summer and (**b**) the EN-1 model for one week in the winter.

According to the results, the functionality of the proposed IMS system in power prediction demonstrated that the amount of experimental dataset is sufficient to train the neural network model for power prediction in a certain period of time.

4.2.2. Fault Detection (FD) Layer

In this section, we introduce a binary classification layer, called Fault Detection (FD), which is a part of the proposed monitoring system. As mentioned previously, we have examined both normal and faulty events in the fault detection (FD) layer and labeled them as the first and second class, respectively. In this regard, we have recorded 433 normal and 1597 faulty events to train and validate the layer. The fault detection is defined as a binary classification problem in which two classes of data are used in the training and validation steps. To evaluate the performance of FD layer, we have recorded 156 faulty and 60 normal events as the unseen dataset. In addition, 1624 training samples (comprised of 342 normal and 1282 faulty events) and 406 validation samples (including 91 normal and 315 faulty events) are chosen randomly to assess the layer. Besides, a comparison is arranged between the trained fault detection layer and a majority voting-based ensemble learning algorithm. For further verification, we have also compared the layer individually to each of the three algorithms presented in this paper.

The fault detection layer is perfectly trained in two previously mentioned scenarios. In addition, we deployed a 10-fold Cross-Validation (CV) algorithm to be reassured about the performance of both feature selection and learning processes. Table 6 shows the best selected features and provides information on the efficiency of each learning algorithm by means of two statistical parameters; average accuracy and standard deviation. In summary, Table 6 depicts that despite a reduction in the number of features, the performance of the second scenario is shown to be more satisfying. Moreover, in both scenarios, the proposed method is proved to be more accurate in comparison to other algorithms. Besides, Figure 15 displays how five different algorithms result on the training dataset in the process of feature selection. According to Figure 15d, it can be easily noticed that the proposed method performs the best with five features and outputs a 95% high accuracy as well as a 1.5% low standard deviation.

Table 6. The training results of the FD layer.

Layer Type	**Scenario 1**					**Scenario 2**				
	Train Accuracy (Standard Deviation)					**Selected Features Train Accuracy (Standard Deviation)**				
									EN	
	NB	**SVM**	**KNN**	**Major Voting**	**The Proposed Method**	**NB**	**SVM**	**KNN**	**Major Voting**	**The Proposed Method**
FD	61.14% (2.2%)	61.7% (3.2%)	86.2% (1.9%)	73.4% (3.1%)	87% (3.2%)	f_1, f_2, f_3, f_4, f_8, f_9 87.7% (3.1%)	f_7, f_8, f_9, f_{11}, f_{16} 92.61% (2.1%)	f_3, f_{13} 94.89% (1.7%)	f_3, f_{16} 92.73% (1.8%)	f_2, f_4, f_5, f_8, f_{16} 96.12% (1.5%)

In the next stage, the algorithms are into the path of being assessed by the validation dataset. Table 7 presents the results of this stage and shows that the proposed method is proved to be more accurate than other algorithms. In Table 7, FD-S1 and FD-S2 signify the process of fault detection in the first and second scenario, respectively.

Finally, each algorithm goes through the process of evaluation via unseen data. The learning algorithms are compared by means of F1-score to show the realistic performance of algorithms in fault classification task. Figure 16 depicts the superiority of F1-score of the proposed method over the other algorithms in this stage.

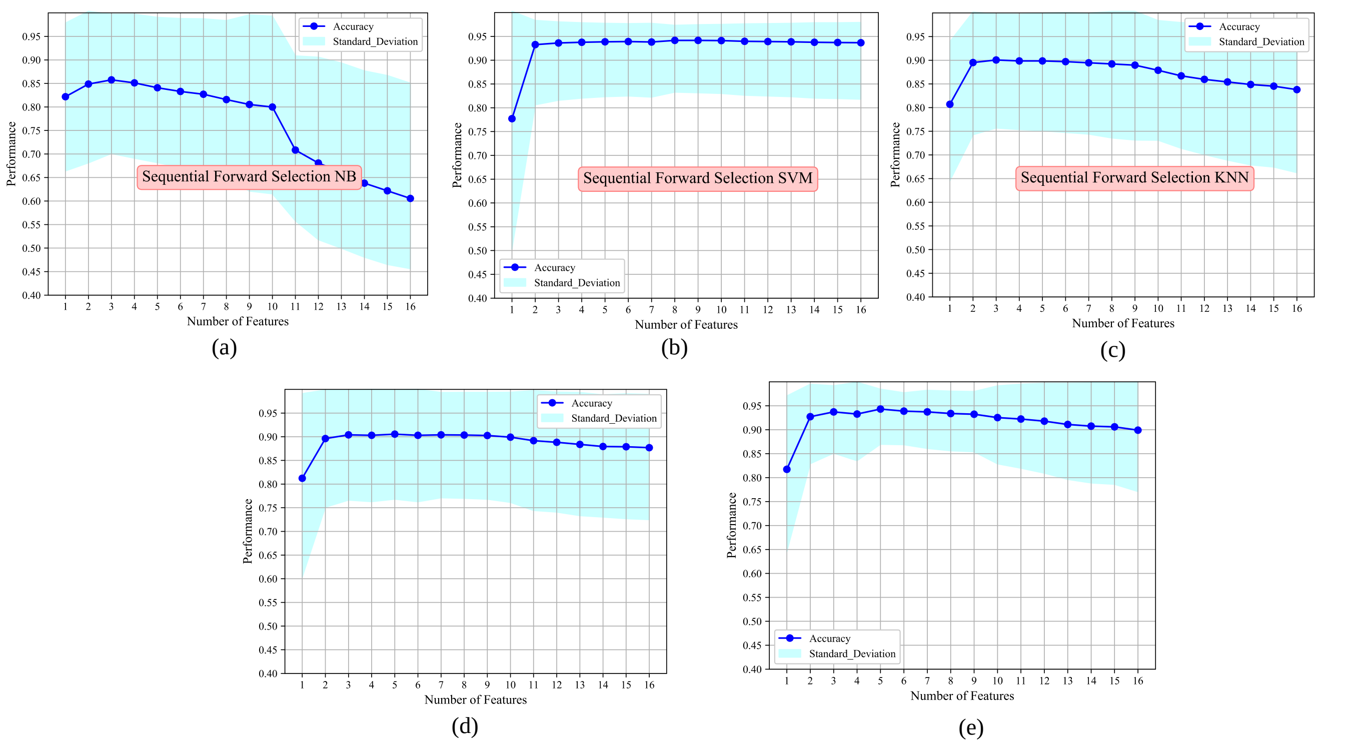


Figure 15. The performance of learning algorithms during the feature selection process in the FD layer, (**a**) NB classifier, (**b**) SVM classifier, (**c**) KNN classifier, (**d**) EN classifier with majority vote, and (**e**) the proposed method.

Table 7. Validation results of the FD layer.

Layer Type	Confusion Matrix (Accuracy %)					$F1_{NB}$	$F1_{SVM}$	$F1_{KNN}$	$F1_{EN}$	
	NB	SVM	KNN	EN: Major Voting	EN: The Proposed Method				Major Voting	The Proposed Method
FD-S1	$\begin{bmatrix} 82 & 9 \\ 160 & 155 \end{bmatrix}$ (58.37%)	$\begin{bmatrix} 74 & 17 \\ 124 & 191 \end{bmatrix}$ (65.27%)	$\begin{bmatrix} 42 & 49 \\ 10 & 305 \end{bmatrix}$ (85.46%)	$\begin{bmatrix} 74 & 17 \\ 100 & 215 \end{bmatrix}$ (71.18%)	$\begin{bmatrix} 69 & 22 \\ 35 & 280 \end{bmatrix}$ (85.96%)	$F1_{(C1)}$ = 96% $F1_{(C2)}$ = 0.96%	$F1_{(C1)}$ =93% $F1_{(C2)}$ =93%	$F1_{(C1)}$ = 96% $F1_{(C2)}$ = 96%	$F1_{(C1)}$ = 96% $F1_{(C2)}$ = 96%	$F1_{(C1)}$ = 98% $F1_{(C2)}$ = 98%
FD-S2	$\begin{bmatrix} 70 & 21 \\ 40 & 275 \end{bmatrix}$ (84.97%)	$\begin{bmatrix} 91 & 0 \\ 33 & 282 \end{bmatrix}$ (91.87%)	$\begin{bmatrix} 87 & 4 \\ 23 & 292 \end{bmatrix}$ (93.34%)	$\begin{bmatrix} 87 & 4 \\ 35 & 280 \end{bmatrix}$ (90.39%)	$\begin{bmatrix} 82 & 9 \\ 13 & 302 \end{bmatrix}$ (94.58%)	$F1_{(C1)}$ = 97% $F1_{(C2)}$ = 97%	$F1_{(C1)}$ = 97% $F1_{(C2)}$ = 97%	$F1_{(C1)}$ = 96% $F1_{(C2)}$ = 96%	$F1_{(C1)}$ = 96% $F1_{(C2)}$ = 96%	$F1_{(C1)}$ = 99% $F1_{(C2)}$ = 100%

4.2.3. Fault Classification (FC) Layer

In this section, we describe the details of a multi-class fault classification layer, developed to detect a wide range of faults on PV strings.

To train a classifier, we need training data and its labels. In this experiment, the labels consist of three classes, namely open-circuit faults, array degradation faults, and string degradation faults. The number of training and validation data is equal to 1597 events, including 433 of the first class, 582 of the second class, and 582 of the third class. In this context, we considered 1277 events for the training purpose (333 events of the first, 472 of the second, and the same of the third class) and 320 events for the validation purpose (100 events of the first, 110 of the second, and the same of the last class). Apart from that, to test the layer accurately, we have provided 156 fault events as unseen data comprising 60 events of the first, 48 of the second, and also 48 of the third class.

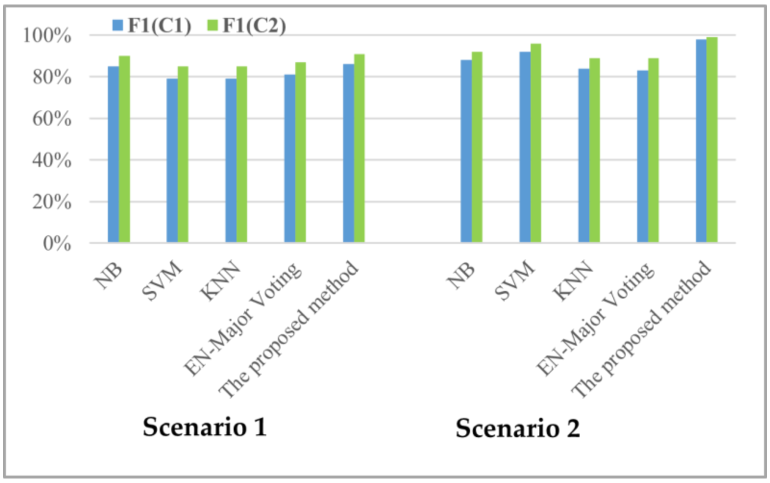


Figure 16. The results of testing the algorithms under two scenarios in the process of fault detection.

Table 8 summarizes the performance of different learning algorithms in the process of fault classification in two scenarios. In the first scenario, according to Table 8, the proposed method is proved to be more accurate (89.11%) than the majority voting-based ensemble learning algorithm (67.34%) in spite of the unsatisfactory performance of SVM and NB algorithms. Besides, in the second scenario, faults are shown to be classified far more accurately by the proposed method with an accuracy of 94.74% and a 2.4% standard deviation.

Table 8. The training results of the FC layer.

Model Type	Scenario 1					Scenario 2				
	Training Accuracy (Standard Deviation)					Selected Features Training Accuracy (Standard Deviation)				
	NB	SVM	KNN	EN: Major Voting	EN: The Proposed Method	NB	SVM	KNN	EN: Major Voting	EN: The Proposed Method
FC	66.79% (4.5%)	63.35% (5.2%)	89.03% (2.2%)	67.34% (4.6%)	89.11% (2.5%)	f_6, f_{11}, f_{13}, f_{14}, f_{16} 75.56% (2.9%)	f_2, f_6, f_9, f_{15}, f_{16} 90% (2.7%)	f_4, f_6, f_{16} 90.9% (3.6%)	f_6, f_9, f_{10}, f_{11}, f_{16} 84.64% (2.9%)	f_2, f_7, f_{16} 94.74% (2.4%)

Figure 17 shows how five different algorithms perform on the training dataset in the process of feature selection. Looking closely at Figure 17d, it is evident that the proposed method performs the best with three features.

Then, we assess the algorithms in two scenarios by the validation dataset in terms of fault classification. As is clearly depicted in Table 9, in the first scenario, the SVM and NB algorithms demonstrate a poorer performance than KNN while the proposed method is shown to be more efficient in fault classification in comparison to the majority voting-based ensemble model. It even exceeds the result of the first scenario and presents a high accuracy of 95.93% in the second scenario, which proves the effectiveness the feature selection algorithm in the accurate classification of faults in PV systems.

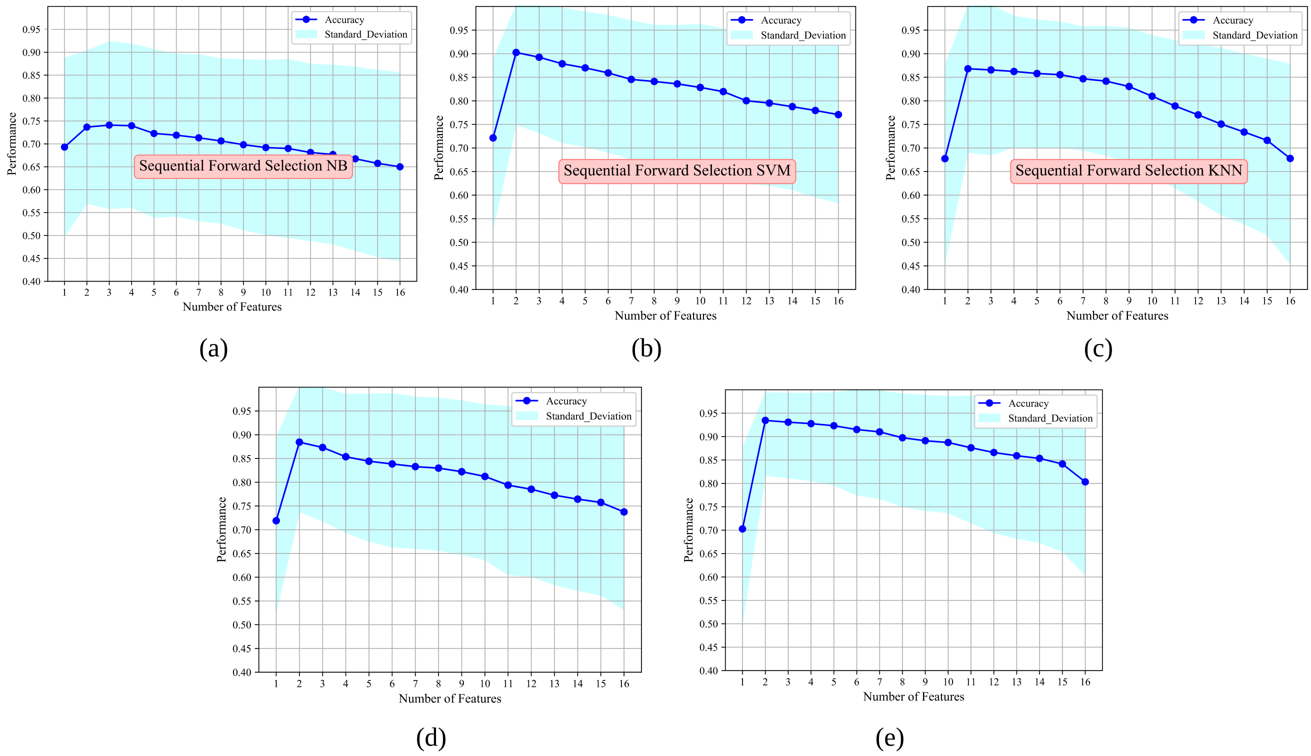


Figure 17. The performance of learning algorithms during the feature selection process in FC layer, (**a**) NB classifier, (**b**) SVM classifier, (**c**) KNN classifier, (**d**) EN classifier with majority vote, and (**e**) the proposed method.

Table 9. Validation results of the FC layer.

Layer Type	Confusion Matrix (Accuracy %)					$F1_{NB}$	$F1_{SVM}$	$F1_{KNN}$	$F1_{EN}$	
	NB	SVM	KNN	EN						
				Major Voting	Proposed Method				Major Voting	Proposed Method
FC-S1	$\begin{bmatrix} 79 & 0 & 21 \\ 32 & 67 & 11 \\ 43 & 4 & 63 \end{bmatrix}$ (65.31%)	$\begin{bmatrix} 84 & 0 & 16 \\ 41 & 62 & 7 \\ 48 & 0 & 62 \end{bmatrix}$ (65%)	$\begin{bmatrix} 93 & 1 & 6 \\ 6 & 99 & 5 \\ 8 & 0 & 102 \end{bmatrix}$ (91.87%)	$\begin{bmatrix} 84 & 0 & 16 \\ 36 & 67 & 7 \\ 43 & 0 & 67 \end{bmatrix}$ (68.12%)	$\begin{bmatrix} 93 & 0 & 7 \\ 7 & 99 & 4 \\ 11 & 0 & 99 \end{bmatrix}$ (90.93%)	$F1_{(C1)}$ =49% $F1_{(C2)}$ =54% $F1_{(C3)}$ =90%	$F1_{(C1)}$ =33% $F1_{(C2)}$ =32% $F1_{(C3)}$ =76%	$F1_{(C1)}$ =92% $F1_{(C2)}$ =67% $F1_{(C3)}$ =92%	$F1_{(C1)}$ =63% $F1_{(C2)}$ =56% $F1_{(C3)}$ =88%	$F1_{(C1)}$ =96% $F1_{(C2)}$ =81% $F1_{(C3)}$ =96%
FC-S2	$\begin{bmatrix} 92 & 0 & 8 \\ 24 & 80 & 6 \\ 32 & 1 & 77 \end{bmatrix}$ (77.81%)	$\begin{bmatrix} 100 & 0 & 0 \\ 11 & 99 & 0 \\ 15 & 0 & 95 \end{bmatrix}$ (91.87%)	$\begin{bmatrix} 100 & 0 & 0 \\ 11 & 99 & 0 \\ 6 & 0 & 104 \end{bmatrix}$ (94.68%)	$\begin{bmatrix} 96 & 4 & 0 \\ 11 & 95 & 4 \\ 21 & 0 & 89 \end{bmatrix}$ (87.5%)	$\begin{bmatrix} 96 & 4 & 0 \\ 9 & 101 & 0 \\ 0 & 0 & 110 \end{bmatrix}$ (95.93%)	$F1_{(C1)}$ =81% $F1_{(C2)}$ =75% $F1_{(C3)}$ =98%	$F1_{(C1)}$ =100% $F1_{(C2)}$ =83% $F1_{(C3)}$ =94%	$F1_{(C1)}$ =79% $F1_{(C2)}$ =76% $F1_{(C3)}$ =97%	$F1_{(C1)}$ =95% $F1_{(C2)}$ =88% $F1_{(C3)}$ =98%	$F1_{(C1)}$ =100% $F1_{(C2)}$ =100% $F1_{(C3)}$ =100%

At last, in order for the proposed method to be fully confirmed, we test the algorithms via unseen data. Figure 18 compares the performance of the algorithms in both scenarios and clearly shows that our proposed method is 100% accurate in the second scenario.

For further validation, the performance of the proposed method is compared with the results of three different methods proposed by other studies. These methods are the most authoritative studies in the detection and classification of faults in PV systems. The comparative results are listed in Table 10. The proposed method has a higher accuracy, and this accuracy is achieved with a very low number of datasets. It should be noted that the presence of high fault impedances leads to low accuracy in fault detection and classification because the operating conditions of the PV system in these faults are close to normal conditions.

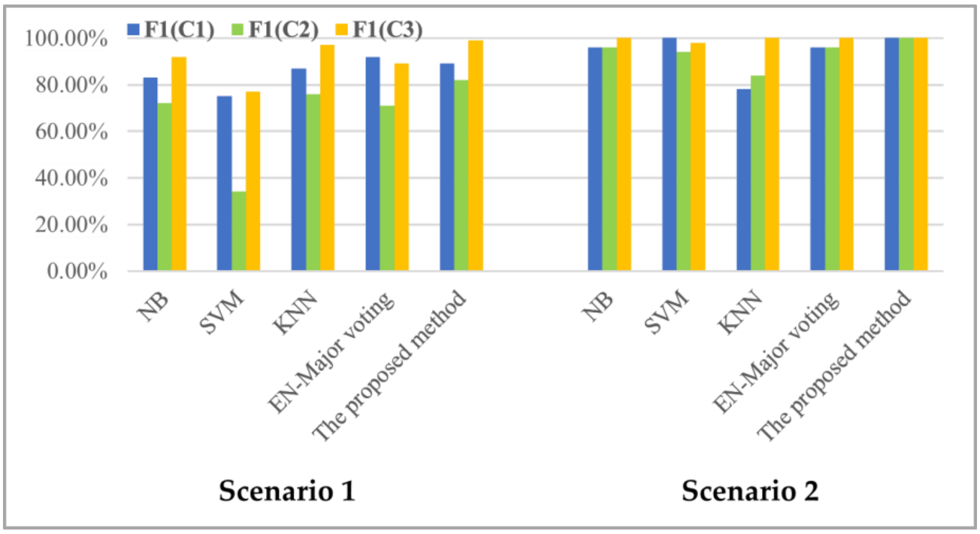


Figure 18. The results of testing the algorithms under two scenarios in the process of fault classification.

Table 10. Comparative results between the proposed method and other studies.

Case Study	Machine Learning Technique	Fault Impedance Range	No. of Learning Datasets Required	Average Accuracy
This study	Ensemble learning algorithm	0–15 Ω	2030	96.72%
[38]	Graph-based semi-supervised learning	0 Ω and 10 Ω	5594	64.61%
[39]	Random Forest algorithm	4 Ω	108,000	98%
[40]	SVM learning algorithm	0–25 Ω	2976	93.07%

5. Conclusions

In this study, an IoT-based monitoring system called Intelligent Monitoring System (IMS) for monitoring of PV plants has been developed. The main objective of IMS was to provide a smart and autonomous monitoring solution using lightweight software and cost-efficient hardware. The IMS used deep ensemble models for fault detection and power prediction in PV systems. The fault diagnostic of IMS was based on the following stages. Firstly, the main features were elicited via analyzing Current–Voltage (I–V) characteristics in several faulty and normal events. Secondly, an ensemble learning model including Naive Bayes (NB), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) was used for detecting and classifying fault events. To obtain a higher performance in diagnosing the faults in PV systems, a feature selection algorithm was also applied. IMS is an interoperable, scalable, and replicable solution for holistic monitoring of PV plants from data acquisition, storing, pre-and post-processing to malfunction and failure diagnosis, performance and energy yield assessment and output power prediction. The system processing can be performed on the cloud server and used Internet of Things (IoT) applications for data transfer and user access. This monitoring solution can be used in remote regions, urban, and rural. The results have shown that the LSTM ensemble network has unique accuracy compared to other models for output prediction, failure identification, and decision-making for remedial actions. In this way, the proposed method was able to detect and classify the faults with an average accuracy of 96.56% and 96.89%, respectively. Moreover, the proposed LSTM ensemble algorithm demonstrated a very small forecasting error compared to the other proposed algorithms.

In future work, we plan to diagnose all the electrical faults including bypass diode problem, line-ground, shading, etc., accurately in PV arrays. For this purpose, we will apply further learning algorithms that can be difficult to choose in ensemble learning models. Therefore, optimization methods such as the genetic algorithm or particle swarm optimization algorithm can be used to select the best learning algorithms. Moreover, we will use different loss functions in order to improve the accuracy in power forecasting.

Author Contributions: Conceptualization, invention, investigation, simulation, visualization, and writing—M.E.; Conceptualization, invention, investigation, simulation, visualization, writing, and editing—A.E.; conceptualization, invention, investigation, validation, visualization, writing, editing, and supervision—M.A.; writing, editing, and visualization—A.N.; writing, editing—A.M.S.; conceptualization, invention, validation, and supervision—J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jager-Waldau, A. Snapshot of Photovoltaics-March 2021. *EPJ Photovolt.* **2021**, *12*, 2. [CrossRef]
2. Daher, D.H.; Gaillard, L.; Ménézo, C. Experimental Assessment of Long-Term Performance Degradation for a PV Power Plant Operating in a Desert Maritime Climate. *Renew. Energy* **2022**, *187*, 44–55. [CrossRef]
3. Aghaei, M.; Fairbrother, A.; Gok, A.; Ahmad, S.; Kazim, S.; Lobato, K.; Oreski, G.; Reinders, A.; Schmitz, J.; Theelen, M. Review of Degradation and Failure Phenomena in Photovoltaic Modules. *Renew. Sustain. Energy Rev.* **2022**, *159*, 112160. [CrossRef]
4. Eskandari, A.; Milimonfared, J.; Aghaei, M. Fault Detection and Classification for Photovoltaic Systems Based on Hierarchical Classification and Machine Learning Technique. *IEEE Trans. Ind. Electron* **2020**, *68*, 12750–12759. [CrossRef]
5. Sizkouhi, A.M.; Esmailifar, S.; Aghaei, M.; Karimkhani, M. RoboPV: An Integrated Software Package for Autonomous Aerial Monitoring of Large Scale PV Plants. *Energy Convers. Manag.* **2022**, *254*, 115217. [CrossRef]
6. Eskandari, A.; Milimonfared, J.; Aghaei, M.; Reinders, A.H. Autonomous Monitoring of Line-to-Line Faults in Photovoltaic Systems by Feature Selection and Parameter Optimization of Support Vector Machine Using Genetic Algorithm. *Appl. Sci.* **2020**, *10*, 5527. [CrossRef]
7. Eskandari, A.; Milimonfared, J.; Aghaei, M.; de Oliveira, A.K.V.; Ruther, R. Line-to-Line Faults Detection for Photovoltaic Arrays Based on I-V Curve Using Pattern Recognition. In Proceedings of the 2019 IEEE 46th Photovoltaic Specialists Conference (PVSC), Chicago, IL, USA, 16–21 June 2019; pp. 0503–0507.
8. Gonzalo, A.P.; Marugán, A.P.; Márquez, F.P.G. Survey of Maintenance Management for Photovoltaic Power Systems. *Renew. Sustain. Energy Rev.* **2020**, *134*, 110347. [CrossRef]
9. Ansari, S.; Ayob, A.; Lipu, M.; Saad, M.; Hussain, A. A Review of Monitoring Technologies for Solar PV Systems Using Data Processing Modules and Transmission Protocols: Progress, Challenges and Prospects. *Sustainability* **2021**, *13*, 8120. [CrossRef]
10. Forero, N.; Hernández, J.; Gordillo, G. Development of a Monitoring System for a PV Solar Plant. *Energy Convers. Manag.* **2006**, *47*, 2329–2336. [CrossRef]
11. Chouder, A.; Silvestre, S.; Taghezouit, B.; Karatepe, E. Monitoring, Modelling and Simulation of PV Systems Using LabVIEW. *Sol. Energy* **2013**, *91*, 337–349. [CrossRef]
12. Tina, G.M.; Grasso, A.D. Remote Monitoring System for Stand-Alone Photovoltaic Power Plants: The Case Study of a PV-Powered Outdoor Refrigerator. *Energy Convers. Manag.* **2014**, *78*, 862–871. [CrossRef]
13. Sutikno, T.; Purnama, H.S.; Pamungkas, A.; Fadlil, A.; Alsofyani, I.M.; Jopri, M.H. Internet of Things-Based Photovoltaics Parameter Monitoring System Using NodeMCU ESP8266. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 5578–5587. [CrossRef]
14. Cheddadi, Y.; Cheddadi, H.; Cheddadi, F.; Errahimi, F.; Es-sbai, N. Design and Implementation of an Intelligent Low-Cost IoT Solution for Energy Monitoring of Photovoltaic Stations. *SN Appl. Sci.* **2020**, *2*, 1165. [CrossRef]
15. Han, J.; Lee, I.; Kim, S.-H. User-Friendly Monitoring System for Residential PV System Based on Low-Cost Power Line Communication. *IEEE Trans. Consum. Electron.* **2015**, *61*, 175–180. [CrossRef]
16. Han, J.; Jeong, J.-D.; Lee, I.; Kim, S.-H. Low-Cost Monitoring of Photovoltaic Systems at Panel Level in Residential Homes Based on Power Line Communication. *IEEE Trans. Consum. Electron.* **2017**, *63*, 435–441. [CrossRef]
17. Touati, F.; Al-Hitmi, M.; Chowdhury, N.A.; Hamad, J.A.; Gonzales, A.J.S.P. Investigation of Solar PV Performance under Doha Weather Using a Customized Measurement and Monitoring System. *Renew. Energy* **2016**, *89*, 564–577. [CrossRef]
18. Pereira, R.I.S.; Dupont, I.M.; Carvalho, P.C.M.; Jucá, S.C.S. IoT Embedded Linux System Based on Raspberry Pi Applied to Real-Time Cloud Monitoring of a Decentralized Photovoltaic Plant. *Measurement* **2018**, *114*, 286–297. [CrossRef]

19. López-Vargas, A.; Fuentes, M.; García, M.V.; Muñoz-Rodríguez, F.J. Low-Cost Datalogger Intended for Remote Monitoring of Solar Photovoltaic Standalone Systems Based on Arduino^{TM}. *IEEE Sens. J.* **2019**, *19*, 4308–4320. [CrossRef]
20. López-Vargas, A.; Fuentes, M.; Vivar, M. IoT Application for Real-Time Monitoring of Solar Home Systems Based on Arduino^{TM} with 3G Connectivity. *IEEE Sens. J.* **2018**, *19*, 679–691. [CrossRef]
21. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; van Deventer, W.; Horan, B.; Stojcevski, A. Forecasting of Photovoltaic Power Generation and Model Optimization: A Review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 912–928. [CrossRef]
22. Samara, S.; Natsheh, E. Intelligent Real-Time Photovoltaic Panel Monitoring System Using Artificial Neural Networks. *IEEE Access* **2019**, *7*, 50287–50299. [CrossRef]
23. Wang, F.; Zhen, Z.; Wang, B.; Mi, Z. Comparative Study on KNN and SVM Based Weather Classification Models for Day Ahead Short Term Solar PV Power Forecasting. *Appl. Sci.* **2017**, *8*, 28. [CrossRef]
24. Raza, M.Q.; Mithulananthan, N.; Li, J.; Lee, K.Y.; Gooi, H.B.; Nadarajah, M. An Ensemble Framework for Day-Ahead Forecast of PV Output Power in Smart Grids. *IEEE Trans. Ind. Inform.* **2018**, *15*, 4624–4634. [CrossRef]
25. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
26. Berghout, T.; Benbouzid, M.; Bentrcia, T.; Ma, X.; Djurović, S.; Mouss, L.-H. Machine Learning-Based Condition Monitoring for PV Systems: State of the Art and Future Prospects. *Energies* **2021**, *14*, 6316. [CrossRef]
27. Basnet, B.; Chun, H.; Bang, J. An Intelligent Fault Detection Model for Fault Detection in Photovoltaic Systems. *J. Sens.* **2020**, *2020*, 6960328. [CrossRef]
28. Dhimish, M.; Holmes, V.; Mehrdadi, B.; Dales, M.; Mather, P. Photovoltaic Fault Detection Algorithm Based on Theoretical Curves Modelling and Fuzzy Classification System. *Energy* **2017**, *140*, 276–290. [CrossRef]
29. Garoudja, E.; Chouder, A.; Kara, K.; Silvestre, S. An Enhanced Machine Learning Based Approach for Failures Detection and Diagnosis of PV Systems. *Energy Convers. Manag.* **2017**, *151*, 496–513. [CrossRef]
30. Appiah, A.Y.; Zhang, X.; Ayawli, B.B.K.; Kyeremeh, F. Review and Performance Evaluation of Photovoltaic Array Fault Detection and Diagnosis Techniques. *Int. J. Photoenergy* **2019**, *2019*, 6953530. [CrossRef]
31. Abdelgayed, T.S.; Morsi, W.G.; Sidhu, T.S. Fault Detection and Classification Based on Co-Training of Semisupervised Machine Learning. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1595–1605. [CrossRef]
32. Harrou, F.; Sun, Y.; Taghezouit, B.; Saidi, A.; Hamlati, M.-E. Reliable Fault Detection and Diagnosis of Photovoltaic Systems Based on Statistical Monitoring Approaches. *Renew. Energy* **2018**, *116*, 22–37. [CrossRef]
33. Eskandari, A.; Milimonfared, J.; Aghaei, M. Line-Line Fault Detection and Classification for Photovoltaic Systems Using Ensemble Learning Model Based on IV Characteristics. *Sol. Energy* **2020**, *211*, 354–365. [CrossRef]
34. Lee, I.; Lee, K. The Internet of Things (IoT): Applications, Investments, and Challenges for Enterprises. *Bus. Horiz.* **2015**, *58*, 431–440. [CrossRef]
35. Mell, P.; Grance, T. The NIST-National Institute of Standars and Technology-Definition of Cloud Computing. *NIST Spec. Publ.* **2011**, 145–800.
36. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
37. Emamian, M.; Milimonfared, J.; Eskandari, A.; Aghaei, M.; Abardeh, R.H.; Vidal de Oliveira, A.K.; Oliveira, A.K.V. Solar Power Forecasting with LSTM Network Ensemble. In Proceedings of the 36th European Photovoltaic Solar Energy Conference and Exhibition, Marseille, France, 9–13 September 2019; Volume 66, pp. 37–39. [CrossRef]
38. Zhao, Y.; Ball, R.; Mosesian, J.; de Palma, J.F.; Lehman, B. Graph-Based Semi-Supervised Learning for Fault Detection and Classification in Solar Photovoltaic Arrays. *IEEE Trans. Power Electron.* **2014**, *30*, 2848–2858. [CrossRef]
39. Chen, Z.; Han, F.; Wu, L.; Yu, J.; Cheng, S.; Lin, P.; Chen, H. Random Forest Based Intelligent Fault Diagnosis for PV Arrays Using Array Voltage and String Currents. *Energy Convers. Manag.* **2018**, *178*, 250–264. [CrossRef]
40. Yi, Z.; Etemadi, A.H. Line-to-Line Fault Detection for Photovoltaic Arrays Based on Multiresolution Signal Decomposition and Two-Stage Support Vector Machine. *IEEE Trans. Ind. Electron.* **2017**, *64*, 8546–8556. [CrossRef]