

Article

An Evaluation of ANN Algorithm Performance for MPPT Energy Harvesting in Solar PV Systems

Md Tahmid Hussain ¹, Adil Sarwar ¹, Mohd Tariq ¹, Shabana Urooj ^{2,*}, Amal BaQais ³
and Md. Alamgir Hossain ^{4,*}

¹ Department of Electrical Engineering, ZHCET, Aligarh Muslim University, Aligarh 202002, India; gk3910@myamu.ac.in (M.T.H.); adil.sarwar@zhcet.ac.in (A.S.); tariq.ee@zhcet.ac.in (M.T.)

² Department of Electrical Engineering, College of Engineering, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

³ Department of Chemistry, College of Science, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; aabaqeis@pnu.edu.sa

⁴ Queensland Micro and Nanotechnology Centre, Griffith University, Nathan, QLD 4111, Australia

* Correspondence: smurooj@pnu.edu.sa (S.U.); mdalamgir.hossain@griffith.edu.au (M.A.H.)

Abstract: In this paper, the Levenberg–Marquardt (LM), Bayesian regularization (BR), resilient backpropagation (RP), gradient descent momentum (GDM), Broyden–Fletcher–Goldfarb–Shanno (BFGS), and scaled conjugate gradient (SCG) algorithms constructed using artificial neural networks (ANN) are applied to the problem of MPPT energy harvesting in solar photovoltaic (PV) systems for the purpose of creating a comparative evaluation of the performance of the six distinct algorithms. The goal of this analysis is to determine which of the six algorithms has the best overall performance. In the study, the performance of managing the training dataset is compared across the algorithms. The maximum power point tracking energy harvesting system is created using the environment of MATLAB or Simulink, and the produced model is examined using the artificial neural network toolkit. A total of 1000 datasets of solar irradiance, temperature, and voltage were used to train the suggested model. The data are split into three categories: training, validation, and testing. Eighty percent of the total data is used for training the model, and the remaining twenty percent is divided equally for testing and validation. According to the results, the regression values of LM, RP, BR, and BFGS are 1, whereas the regression values for SCG and GDM are less than 1. The gradient values for LM, RP, BFGS, SCG, BR, and GDM are 7.983×10^{-6} , 0.033415 , 1.0211×10^{-7} , 0.14161 , 0.00010493 , and 11.485 , respectively. Similarly, the performance values for these algorithms are 2.0816×10^{-10} , 2.8668×10^{-6} , 9.98×10^{-17} , 0.052985 , 1.583×10^{-7} , and 0.15378 . Overall, the results demonstrate that the LM and BFGS algorithms exhibit superior performance in terms of gradient and overall performance. The RP and BR algorithms also perform well across various metrics, while the SCG and GDM algorithms show comparatively less effectiveness in addressing the proposed problem. These findings provide valuable insights into the relative performance of the six evaluated algorithms for MPPT energy harvesting in solar PV systems.

Keywords: artificial neural network (ANN); solar photovoltaic (PV); maximum power point tracking; Levenberg–Marquardt (LM); Bayesian regularization (BR); scaled conjugate gradient (SCG); resilient backpropagation (RP)



Citation: Hussain, M.T.; Sarwar, A.; Tariq, M.; Urooj, S.; BaQais, A.; Hossain, M.A. An Evaluation of ANN Algorithm Performance for MPPT Energy Harvesting in Solar PV Systems. *Sustainability* **2023**, *15*, 11144. <https://doi.org/10.3390/su151411144>

Academic Editor: Mohamed A. Mohamed

Received: 30 April 2023

Revised: 1 June 2023

Accepted: 12 June 2023

Published: 17 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Our energy usage includes solar PV electricity, which is also a crucial element of renewable energy networks. The cost of PV modules is falling as technology advances quickly, and PV panels are becoming more reliable. National economies are investing heavily in off-grid and grid-connected PV networks [1,2]. PV electricity is unstable and dependent on solar radiation as well as other meteorological conditions, such as humidity, wind speed direction, cloud cover temperature, and precipitation. This makes it different from typical

energy production methods [3]. Power networks have had significant issues because of the advent of large-scale grid-connected solar PV facilities, including a lack of device flexibility, energy balance, and efficiency [4]. For PV networks to have a consistent electricity supply, it is essential to anticipate the solar output energy. The accuracy of the predictive models boosts the device dependability, decreases the cost of extra equipment maintenance, and limits the impact of solar PV performance [5]. Irradiation and temperature are features of a PV module's I-V properties. Solar cell arrays are followed by MPPT controllers for the best utilization performance. A thorough list of 40 distinct MPPT approaches and their categorization was developed in [6]. Several MPPT algorithms and designs are covered in a number of publications in the literature to improve the performance of PV device [7]. The most efficient and widely used methods are perturb and observe (P&O) [8], incremental conductance (INC) [9], fuzzy logic controller (FLC) [10], a P&O technique based on particle swarm optimization (PSO) [11], and ANN [12]. For each of these options, there are differences in the pace of convergence, oscillation across the absolute maximum power point (MPP) complexity, stability, cost, and necessary electrical equipment [13].

After a quick irradiance change that causes a distortion in the algorithm of P&O and the working parameters of PV systems, the controller struggles at first to surpass the MPP [14]. The controller nonetheless mitigates the mistakes of the algorithm, which, with some lag time, follows the MPP once again. Furthermore, the MPP's terminal voltage fluctuates as a consequence amid a power outage. The smallest possible disruption phase size is used to counteract these oscillations. The minor phase again minimizes the transient startup and changes the system sensitivity to the weather. Inc. algorithm; it works well to use a controller such as proportional integral (PI) to deal with quick changes in irradiance and corresponding declines in the oscillation of the MPP during rips. Hence, the response and, according to the INC method, swing pace would still be balanced, but often breaks away from the MPP with sudden irradiance shifts. Authors in [15] claim that these algorithms cannot quickly and accurately detect the entire power because there are oscillations at the highest point. The biological neural networks of the human brain are the motivation for the ANN system development. It is used to train and assess the PV system I-V and P-V nonlinearity relationship. ANN retrieves inputs such as the input voltage, input current, temperature, irradiance, and metrological information and continuously learns to modify how the solar power system behaves for the highest impact [16]. The design of an FLC model is possible using ANN for a more accurate and simple converter actualization [17]. The dataset comes from simply using a simulation or hardware configuration and also introducing solar irradiances, temperatures, and/or the voltage or current of a solar power system to ANN to determine the necessary maximum power (Pmax) or maximum output voltage (Vmax). This information is converted into training data that is provided to the system to teach the desired ANN how to operate. Test datasets are used after training to evaluate the built-in ANN's performance, and for further correction, errors are sent to the ANN [18], which has the capability of predicting MPP by state estimation and SMC filtering (also known as sequential Monte Carlo). The framework of the incremental conductance maximum power point tracking approach (IC MPPT) might be expanded to include a model of state space estimations for successive maximum power points (MPP). The voltage or current and irradiance statistics are used to predict the global MPP (GMPP) in the ANN model to enhance the SMC estimation [19]. Among the advantages of ANN are exceptional precision in modelling and the ability to resolve nonlinearity issues without prior knowledge or models [20]. To speed up and improve tracking through solar power system modelling and forecasting, ANN may be used [21]. It has been demonstrated to have a quicker response time and less oscillation than MPP [22]. In actual operational circumstances, MPPT based on ANN can monitor MPP with little effort. Low ripple and transient time [23]. The square error method is used in the error calculation as a feedback correction [24]. A proper, precise, and systematic training set of data, however, is an important constraint to function well with the ANN without much training error [25]. Yet, the variations in instruction and operation when creating an ANN model and solar system

settings make the training technique challenging. The authors in [26] thus recommended using a particle swarm optimization (PSO) model in MATLAB and Simulink to determine the best starting weights for ANN models by selecting the best topology to improve the accuracy of the ANN model. Hence, with the conflict processing speed and the most accurate regression after resolution of the ANN model, the minimization of the mean squared error occurs. The results show that using the real-world approach of the improved feedforward ANN depends on data from the PSO method; the peak power is accurately predicted with average hourly efficiencies of over 99.67% on bright and 99.30% on overcast days, respectively. An ANN-based MPPT controller demonstrates reduced steady-state error and a quicker response to abrupt changes in solar temperature and irradiance in contrast to both P&O and IC [27]. Nevertheless, an improved algorithm of P&O with variable step size aims to improve the tracking speed and reduce the steady-state fluctuation or oscillation under abrupt changes in irradiance or partial shading conditions (PSC). Integrating FLC and ANN with more established MPPT methods such as IC and P&O is a good fit. The ANN method calculates the MPP even when there is no shade or temperature information available from the panel; nevertheless, the hill-climbing (HC) methodology further enhances the outcome. IC-ANN and P&O-ANN are two more hybrid MPPTs that are linked with the stacked autoencoder (SAE) controller via the use of building blocks and deep learning (DL) training. It is recommended to use a greedy layer-wise method to harvest the maximum amount of energy possible from the solar energy system. Backpropagation and supervised learning are then used to fine-tune the deep neural network using traditional MPPT-IC and P&O. This allows for the greatest amount of power to be extracted [28].

In the literature study, different artificial intelligence (AI) algorithms for energy harvesting with the PV system are highlighted; however, only a few studies have made use of LM algorithms, BR algorithms, SCG algorithms, etc. There is reason to be optimistic about the ability of AI systems to anticipate optimum power with low error under a variety of meteorological situations. Processing vast volumes of data is achieved much more speedily and effectively via the use of neural networks [29]. Even though there has been some study on contrasting various MPPT topologies for solar PV systems, there is still a significant research gap when it comes to evaluating the performance of different ANN algorithm-based MPPT methods for solar energy harvesting. In addition to evaluating the performance of ANN algorithms for MPPT energy harvesting in solar PV systems, the study proposed in [30] also aims to contribute to the literature by introducing a new model predictive control method for buck–boost inverter-based photovoltaic systems. By incorporating this study into the proposed study, we expand the scope and relevance of our research. The proposed model predictive control method offers a novel approach to optimize the operation of buck–boost inverters in photovoltaic systems, thereby enhancing the energy harvesting efficiency. This integration of the new control method into the evaluation of ANN algorithms provides a comprehensive analysis of advanced techniques for improving the performance of solar PV systems. Similarly, the study presented in [31] aims to contribute to the literature by considering the topic of optimal control of an energy-storage system in a microgrid for reducing wind-power fluctuations. By incorporating this work into the study, we can also broaden the scope of our research and address the challenges associated with integrating wind power into microgrid systems. The optimal control of an energy-storage system plays a crucial role in mitigating the intermittent nature of wind power generation and ensuring a stable and reliable energy supply. This integration of the optimal control strategy into the evaluation of ANN algorithms provides a comprehensive analysis of advanced techniques for improving the stability and efficiency of renewable energy systems. By considering both solar and wind power aspects, our study offers insights into the integration of multiple renewable energy sources and their control mechanisms, thereby contributing to the overall understanding of sustainable energy systems.

It is also crucial to investigate alternate methods for training neuro-fuzzy systems in addition to analyzing the effectiveness of different ANN algorithms for MPPT energy

harvesting in solar PV systems. A notable topic in this context is the training of neuro-fuzzy systems using meta-heuristic algorithms presented in [32]. These algorithms offer a promising approach to optimize the parameters of neuro-fuzzy models and enhance their MPPT capabilities. By incorporating meta-heuristic algorithms into the training process, such as genetic algorithms, particle swarm optimization, or simulated annealing, the neuro-fuzzy models can effectively learn the mapping between input data and optimal power outputs. This integration of meta-heuristic algorithms with neuro-fuzzy systems has the potential to improve the accuracy and efficiency of MPPT algorithms, ultimately leading to enhanced energy harvesting in solar PV systems. Therefore, investigating the training of neuro-fuzzy systems using meta-heuristic algorithms presents an intriguing avenue for further research and advancement in the field of MPPT energy harvesting. Furthermore, recent advancements in cooperative optimization techniques for assessing the performance of various ANN algorithms for MPPT energy harvesting in solar PV systems should also be considered. One such notable topic is the improved cooperative artificial neural network–particle swarm optimization (ANN-PSO) approach for solar photovoltaic systems with maximum power point tracking (MPPT) [33]. This approach combines the power of artificial neural networks and particle swarm optimization to enhance the efficiency and accuracy of MPPT algorithms in solar PV systems. By leveraging the cooperative nature of ANN-PSO, the system can benefit from the collective intelligence of multiple agents working together to find the optimal power output. This cooperative approach offers a promising solution for addressing the challenges of MPPT in solar PV systems, including non-linearity, partial shading, and varying environmental conditions. Therefore, incorporating the improved cooperative ANN-PSO approach in the evaluation of ANN algorithms for MPPT energy harvesting would provide valuable insights into its effectiveness and potential as an advanced optimization technique in solar PV systems.

In [34], a comparative analysis of three algorithm is proposed for energy harvesting of solar PV. The proposed work is compared in terms of the performance of handling the trained dataset, and the authors have described the algorithms in a clear and detailed way.

However, this work provides a detailed performance comparison of six different ANN-based algorithms (LM, BR, RP, GDM, BFGS, and SCG) for MPPT solar energy harvesting and also provides an explanation of each algorithm. Recent research on ANN-based MPPT has solely concentrated on fewer approaches. The created model gives a good grasp of how practical and applicable these algorithms are. It also contributes significantly to the existing literature in several ways. Firstly, it provides a comprehensive evaluation of six different ANN algorithms, specifically in the context of MPPT for solar PV systems. By comparing the performance of these algorithms, the study offers valuable insights into their effectiveness, convergence properties, and accuracy. This comparative analysis helps researchers and practitioners make informed decisions when selecting the most suitable algorithm for MPPT in solar PV systems. Additionally, the study incorporates real-time data on the solar irradiance, panel temperature, and generated voltage for training the ANN algorithms. This aspect enhances the practical relevance of the research findings and increases their applicability to real-world scenarios. The methodology and dataset used in this study can serve as a valuable resource for further research and algorithm development in the field. Furthermore, the study introduces and evaluates various performance metrics, such as regression, error at the middle bin, gradient, performance, momentum parameter, and epochs. These metrics provide a comprehensive framework for assessing the efficacy of ANN algorithms in MPPT energy harvesting. By establishing these metrics, the study contributes to the existing literature by providing a standardized approach to evaluate and compare the performance of different algorithms. Overall, this study advances the understanding of ANN algorithm performance for MPPT energy harvesting in solar PV systems and provides valuable insights and guidance for researchers, engineers, and practitioners working in this field. It sets a foundation for further research and development of advanced algorithms and methodologies for optimizing energy harvesting efficiency in solar PV systems. The performance of the six ANN algorithms is evaluated using a

thorough method, which includes training, validation, and testing using generated data of the solar irradiance, temperature, and produced voltage. This method is used to determine how well the algorithm functions. The performance of these algorithms is analyzed via the use of a simulated model that is deployed on MATLAB/Simulink. This offers clear knowledge of the application of ANN algorithms for MPPT in solar PV systems. The (ANN)-based MPPT algorithm is trained using data taken from the actual world, proving both the usefulness and efficacy of this technology.

The article's remaining sections are organized as follows: the status of ANN and MPPT technology is discussed in Section 2. Section 3 describes the modelling of ANN-based MPPT for solar PV system. The results and comments for six training algorithms are shown and discussed in Section 4. Lastly, the work is concluded in Section 5.

2. The State of the Art of ANNs and MPPTs

Artificial neural networks, often known as ANNs, are a specific form of machine learning algorithm that attempt to replicate the function and structure of the biological neural networks found in the human brain [35]. They are designed to connect various parameters to particular data points without the need for mathematical equations or complex mathematical bases [36]. In order to train ANNs, a technique known as supervised learning is applied. In this approach, datasets consisting of input–output parameter values are utilized in order to train the network. The datasets are often divided into two groups: a training dataset and a validation dataset. The training dataset is used to train the network, and the validation dataset is used to assess how well the trained network performed. An artificial neural network (ANN) is made up of many different neurons, all of which are linked together by a fractional number that is referred to as weight [37]. In order to accurately forecast the results of the process, the weights are changed while it is in the training phase. The weights become constant once the error falls below a permissible value. The most common form of ANN is the two-layer model, which is shown in Figure 1.

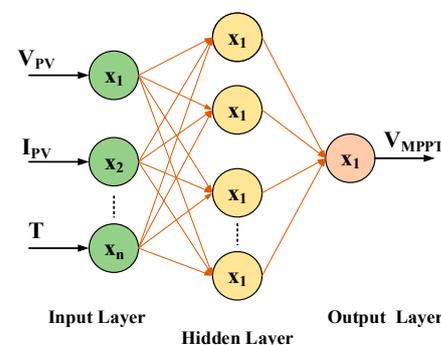


Figure 1. The fundamental design of a two-layer ANN.

The network's inputs are incorporated into the network at various times. The training dataset is used to create the neural network, and the validation dataset is used to assess how well it performed. Once the network has been trained and the error is within permissible limits [38], the validation dataset parameters of input are imported, and the associated values of output parameter are predicted [39]. This process helps to ensure that the ANN is accurate and reliable.

By contrasting the anticipated output parameter values from the validation dataset with the corresponding actual values, the trained ANN's performance is assessed. The trained ANN may be regarded as the best prediction model if the difference between the predicted and actual values is less than the allowable maximum. For the selected training method and number of training iterations, the ANN can forecast the matching input values and output parameters.

The trained ANN along with the training procedure is selected as the optimized model if the error magnitude is smaller than the allowed value. However, if the error is still high,

different training algorithms or more training iterations may be tried before an allowable error is obtained [40]. The results obtained from the validation dataset using the optimal ANN model validate the generalization of the trained ANN [41].

MPPT in Solar PV systems is one of the many power electronics applications where ANNs have found widespread usage. By changing the operating point to the MPP under various environmental circumstances, MPPT aims to maximize the power output from a PV system. The MPP of a PV system is a nonlinear and time-varying function that is difficult to model analytically. ANNs have been shown to be effective tools for modeling this nonlinear relationship, making them a popular choice for MPPT in PV systems.

For MPPT in PV systems, a variety of ANN algorithms, such as radial basis function (RBF) and multi-layer perceptron (MLP) networks, may be utilized. A sort of feedforward neural network called an MLP network is made up of layers of linked neurons. They are commonly used for supervised learning tasks, such as function approximation and regression. One of the most common ANN-based MPPT techniques is the Perturb and Observe (P&O) method with an MLP network. In this method, the PV system operating point is perturbed and the change in power is observed. The change in power and the operating point are used as inputs to the ANN, which is trained to predict the MPP. The ANN output is then used to adjust the operating point to the MPP. This process is repeated in real time to track the MPP as it changes due to varying environmental conditions. Another popular ANN-based MPPT technique is the incremental conductance (IC) method with an RBF network. In this method, the PV system operating point is adjusted incrementally, and the change in power is used as an input to the ANN. The ANN is trained to predict the direction of the MPP, and the operating point is adjusted in the direction predicted by the ANN. This process is also repeated in real time to track the MPP as it changes. There are also ANN-based MPPT techniques that combine multiple algorithms, such as the hybrid MPPT algorithm that uses both the P&O and IC methods. In this method, the P&O method is used to quickly find the initial MPP, and the IC method is used to track the MPP as it changes. One of the main advantages of ANN-based MPPT techniques is that they can effectively model the nonlinear relationship between the PV system output power and the operating conditions. They can also adapt to changing environmental conditions in real time, which is essential for efficient MPPT in PV systems. Additionally, ANNs are relatively easy to implement and can be used with a variety of PV systems, from small-scale systems to large-scale power plants. The fact that ANN-based MPPT approaches need a lot of data to train the network properly is one of their key drawbacks. This can be a significant obstacle for some applications, particularly those that are remote or have limited data collection capabilities. Additionally, the training procedure may also be time-consuming and computationally demanding, which can be problematic for real-time applications.

A variety of maximum power point tracking (MPPT) algorithms for photovoltaic (PV) systems were examined and classified in a study that was referenced in [42]. The classifications were based on the number of control variables and the types of control strategies. In order to examine the dynamic response of the PV voltage ripple, the authors made use of MATLAB/Simulink and the dSPACE framework. They provided a hands-on assessment for commonly used MPPT algorithms [43] and compared them with a PI controller. The research modified environmental variables, including changing irradiance and increasing temperatures, to emphasize the benefits and drawbacks of the P&O and INC algorithms in simulated findings. To improve the P&O algorithm in the context of unforeseen irradiance variations, they put up a novel method [44], which includes two algorithms: an original disturbance algorithm and an adaptive control algorithm. Experimental findings were also compared with the proposed algorithm and traditional algorithms. In another study [45], the authors proposed an updated P&O algorithm to address the root cause of the drift phenomenon and compared the results of experiments and simulations using conventional P&O methods using adaptive measures.

The INC MPPT algorithm has been evaluated in [45]. The authors tested the INC algorithm using an isolated PV pumping approach, which impacted service speeds and altered the reference voltage. Here, the influence of prior disturbances on the phase size and disturbance size has been made clear, which often reveals the algorithm's uncertainty as a result of sudden changes in irradiance. The accomplishment also has something to do with the recommended algorithm.

To monitor the PV properties of MPP, a comparison study is performed between the P&O method and the INC algorithm. In contrast, the transient solution for the INC job ratio disturbance and reference voltage disturbance is more quickly implemented. It was shown that the INC MPPT technique is less susceptible to device dynamics and noise. Greater stability at rapidly changing irradiance has been shown by this phenomenon in the suggested device. The authors of [26] have written up experimental studies with high INC algorithm perturbation rates. At greater disruption speeds, INC is shown to provide a quicker transitional reaction. The method also offers a speedier MPP recovery when the MPP is affected by noise or brightness fluctuations. There have been several MPPT algorithms using PI controllers used so far. However, the most uncomfortable and unusual idea for implementing PV systems was using ANN as the MPPT controllers.

The ANN MPPT was characterized as having a number of off-line preparation characteristics, including nonlinear mapping, a faster response time, and less computing effort in [46]. A novel neural network (NN) MPPT controller for PV systems was proposed by the authors in [47]. Using MATLAB and Simulink, data were extracted from the P&O system to be used in the training and testing of the NN model. The simulation results showed that using the suggested NN controllers for swiftly moving insulation will increase monitoring accuracy, response time, and control. In order to accomplish the electronic power grids' optimal design using AI, the research is important because it integrates configuration parameters with reliability measurements [48]. This article explains how switching the frequency and voltage series may be used to determine a device's stability, efficiency, and cost. Additionally, it provides a thorough examination of the data extraction and training for artificial neural networks. The findings of another ANN MPPT [49] are superior to those of the climbing algorithms.

To locate the global peak using the MATLAB NN-Tool, the authors created a feedback network using the LM backpropagation method. The simulation findings demonstrate that the suggested model is effective and has a smaller root mean square error (RMSE) than the climbing method. The fundamental nature and purpose of diverse MPPTs are achieved using modified methodologies. It is recommended that a technique employ a NN controller instead of a PI controller when utilizing conventional MPPT algorithms. PI-dependent algorithms may thus provide better dynamic stability with abrupt changes in the environment.

3. Modelling of ANN-Based MPPT for Solar PV System

3.1. Solar Photovoltaic (PV) System

A solar photovoltaic (PV) cell, also known as a solar cell, is a device that uses a process known as the photovoltaic effect to transform light energy into electrical energy. The photovoltaic effect occurs when photons from sunlight knock electrons into a higher state of energy, allowing them to flow as an electrical current. Solar PV cells are made of semiconductor materials, typically silicon, and are designed to capture the energy from sunlight and convert it into usable electrical power.

Solar PV cells are connected together to form a solar panel, which can then be connected in series to form a solar array. The efficiency of solar PV cells, or the percentage of sunlight energy that is transformed into electrical energy, varies, but most commercial solar cells have an efficiency of around 15–20%. The performance of a solar cell is also influenced by temperature, shading, and other environmental factors.

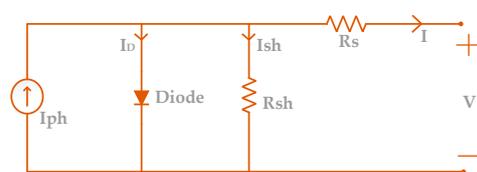
The equivalent circuit of a solar cell is a mathematical model that represents the behavior of a solar cell as an electrical circuit. This model is used to analyze and understand the performance of a solar cell, including its voltage, current, and power output.

A current source, a series resistance (denoted by R_s), a shunt resistance (denoted by R_{sh}), and a diode are the standard components that make up the equivalent circuit of a solar cell. The current source is symbolic of the photocurrent that is produced by the solar cell, and the R_s is symbolic of the resistance that is posed by the material and connections that are included inside the cell. The R_{sh} represents the resistance to current flow through pathways other than the intended path, such as cracks or defects in the cell. The diode represents the non-linear behavior of the solar cell, including its short-circuit current and open-circuit voltage.

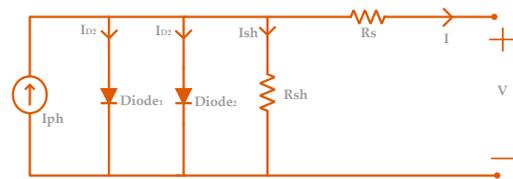
By analyzing the equivalent circuit, one can better understand the factors that influence the performance of a solar cell and make design improvements to increase its efficiency. For example, reducing the values of R_s and R_{sh} can increase the overall performance of the solar cell. Additionally, the equivalent circuit can be used to determine the MPP of a solar cell, which is the point at which the solar cell produces the maximum amount of power.

There are several different types of equivalent circuits used to model the behavior of solar cells, each with its own advantages and limitations. The most common types of equivalent circuits are:

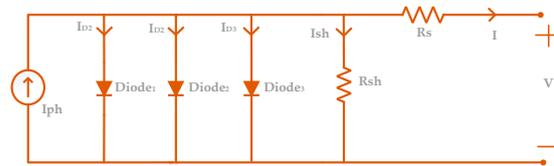
1. One-diode model (Scheme 1): The one-diode model is the simplest type of equivalent circuit and is often used as a first approximation of the behavior of a solar cell. It consists of only a current source and a diode and is relatively easy to analyze and understand. The main advantage of the one-diode model is its simplicity, which makes it suitable for many applications where a quick estimate of the performance of a solar cell is needed.
2. Two-diode model (Scheme 2): The two-diode model is a more sophisticated equivalent circuit that includes two diodes and a series resistance. This model is used to represent the behavior of a solar cell more accurately and is particularly useful for analyzing the performance of cells under varying light and temperature conditions. The main advantage of the two-diode model is its improved accuracy compared to the one-diode model, which makes it suitable for applications where a more detailed understanding of the behavior of a solar cell is needed.
3. Circular model: The circular model is an advanced equivalent circuit that includes a series resistance, a shunt resistance, and a diode. This model represents the behavior of a solar cell more accurately than the one-diode or two-diode models and is particularly useful for analyzing the behavior of cells under complex environmental conditions. The main advantage of the circular model is its improved accuracy and the ability to model the effects of shading and other environmental factors on the performance of a solar cell.
4. Three-diode model (Scheme 3): It is a more sophisticated equivalent circuit that consists of three diodes and a series resistance. This model is used to represent the behavior of a solar cell more accurately, particularly under conditions of high light intensity and high temperature. The three-diode model takes into account the non-linear behavior of a solar cell under these conditions and provides a more accurate representation of the behavior of a solar cell than the circular model or the one- or two-diode models.



Scheme 1. Equivalent circuit of one diode model.



Scheme 2. Equivalent circuit of two diode model.



Scheme 3. Equivalent circuit of three diode model.

A simulation model for a photovoltaic (PV) cell is created to optimize its power conversion efficiency. This is achieved by taking into account the impact of light intensity and temperature on the cell’s production capacity. The analogous electrical circuit of the PV cell, as displayed in Figure 2, is used as the basis for this simulation model. This permits the measurement of the maximum power point of the solar panels and provides a more accurate forecast of the performance of the cell under various environmental circumstances. Additionally, this makes it possible to determine the maximum power point.

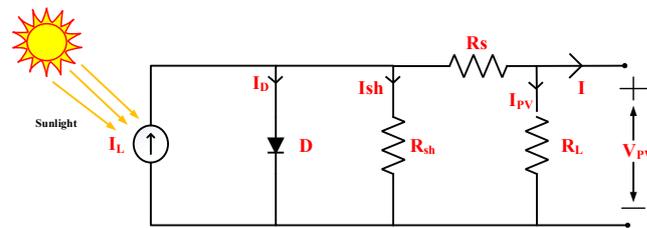


Figure 2. Analogous circuit of a solar PV cell.

In Figure 3a, the current and voltage for a photovoltaic cell are shown in response to various levels of irradiance, including (200 W/m², 400 W/m², 600 W/m², 800 W/m², and 1000 W/m²) at 25 °C. The voltage and current of the cell fluctuate as a consequence of variations in the sun irradiation.

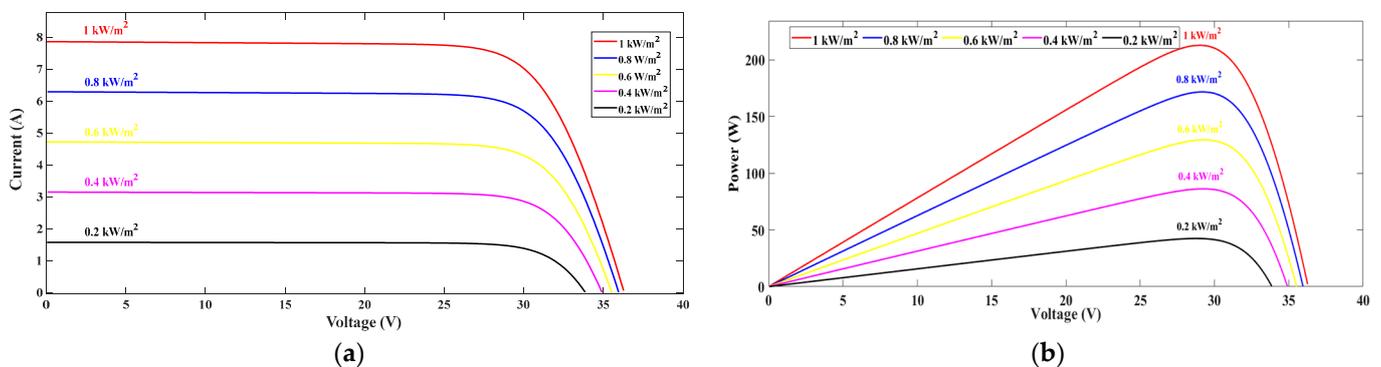


Figure 3. Characteristic curve of module 1Soltech ISTH-215-P: (a) I-V curve; (b) P-V curve.

Figure 3b demonstrates the impact of light intensity on the production capacity of the photovoltaic cell. It shows that even with the same temperature, the optimal energy point for the cell occurs at a constant level of light intensity.

The solar cell current can be expressed using the variables such as short circuit current (I_{sc}), saturation current (I_0), diode ideality constant (a), number of series-connected cells

(N_s), temperature of the cell (T), Boltzmann constant (K), charge of an electron (q), series resistance (R_s), and shunt resistance (R_{sh}) of the array. These variables are used in Equations (1) and (2) to represent the solar cell current [50].

$$I = I_{sc} - I_0 \left(e^{\frac{V+IR_s}{N_sKT}} - 1 \right) - \frac{V + IR_s}{R_{sh}} \quad (1)$$

$$I = I_{sc} - I_0 \left(e^{\frac{V+IR_s}{V_T}} - 1 \right) - \frac{V + IR_s}{R_{sh}} \quad (2)$$

where the value of K and q are 1.38×10^{-23} J/K and 1.6×10^{-19} C, respectively. Equation (1) can be rewritten as Equation (2), if the array thermal voltage is replaced by $V_T = \frac{N_sKT}{q}$.

3.2. ANN-Based MPPT for Solar PV System

The MATLAB/Simulink—based simulated model of the solar PV system, illustrated in Figure 4, has been developed. It utilizes the 1Soltech ISTH-215-P solar panel, with the electrical specifications provided in Table 1. The simulated model consists of two primary subsystems: the ANN_MPPT and Switching block. The proposed system incorporates an ANN algorithm, which is represented by an ANN block as depicted in Figure 5.

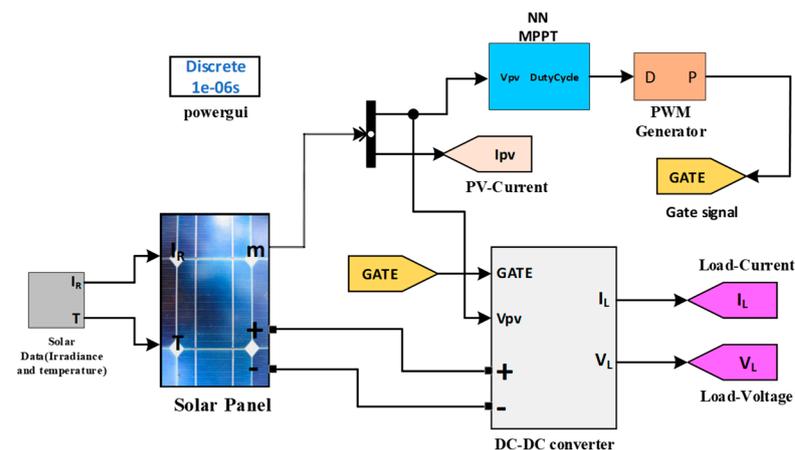


Figure 4. ANN-Based MPPT Solar PV Model.

Table 1. Specification of 1Soltech ISTH-215-P Solar Module.

Specification	Value
Power at STC (W)	215
Power at PTC (W)	189.4
V_{mp} : Voltage at Max Power (V)	29.0
I_{mp} : Current at Max power (A)	7.35
V_{oc} : Open Circuit Voltage (V)	36.3
I_{sc} : Short Circuit Current (A)	7.84
Nominal Operating Cell Temperature ($^{\circ}$ C)	47.4
Open Circuit Voltage Temp Coefficient ($\%/^{\circ}$ C)	−0.361
Short Circuit Current Temp Coefficient ($\%/^{\circ}$ C)	0.102
Max power Temp Coefficient ($\%/^{\circ}$ C)	−0.495
Power Density at STC (W/m^2)	136.943
Power Density at PTC (W/m^2)	120.637

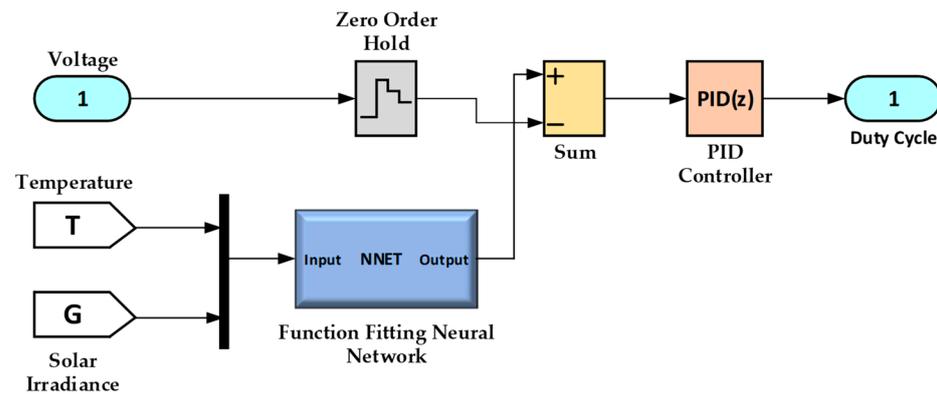


Figure 5. Proposed Block Diagram of ANN Algorithm.

Within the ANN_MPPT subsystem, a comparator compares the output voltage, V_1 , of the ANN with the voltage generated by the PV array. This generated voltage serves as the reference for the comparator. A Proportional Integral Derivative (PID) controller generates a duty cycle signal based on the difference between V and V_1 . The Switching block incorporates a boost converter where the insulated-gate bipolar transistor (IGBT) is activated by the gate signal produced by the pulse width modulated (PWM) generator. The duty cycle of PWM is controlled by the voltage difference observed by the comparator. The ANN algorithm ensures a consistent duty cycle for PWM through the ideal correlation between the target and trained values, leading to smooth IGBT switching operations. Additionally, the solar data subsystem sequentially provides input data (irradiance and array temperature) to the PV array, ensuring that the simulation time corresponds to the time taken for transferring the input data.

Solar radiation intensity and panel temperature both have an impact on how much power can be generated by solar panels. Hence, the input data for the artificial neural network (ANN) model is computed depends on the solar irradiance and temperature using Formulas (3) and (4).

Irradiance Calculation, G (W/m^2):

$$G = [(G_{max} - G_{min}) \times rand] + G_{min} \quad (3)$$

Temperature Calculation, T ($^{\circ}\text{C}$):

$$T = [(T_{max} - T_{min}) \times rand] + T_{min} \quad (4)$$

Maximum Voltage (V_{MP}), at given G and T

$$V_{MP} = V_{OC} + (beta \times (T - T_S)) \quad (5)$$

where V_{OC} is open circuit voltage of panel and T_S is standard temperature.

Maximum Current (I_{MP}) at given G and T

$$I_{MP} = I_M \times \left(\frac{G}{G_S}\right) \times (1 + (alpha \times (T - T_S))) \quad (6)$$

Maximum Power (P_{MP}) at given G and T

$$P_{MP} = I_{MP} \times V_{MP} \quad (7)$$

The MPPT technology that is used for the solar panel system is built with the help of data on solar irradiation, the maximum voltage produced and temperature. The solar PV array receives specific data about the amount of solar irradiation as well as the temperature from the solar data system. Equations (5)–(7) are used in order to compute the solar panel

maximum produced power (P_{MP}), maximum generated voltage (V_{MP}), and maximum generated current (I_{MP}) for a variety of irradiance and temperature conditions. The data that are considered to be the output of the neural network are the voltage that is produced by the solar panel.

The rated current and temperature coefficient and rated voltage of the solar panel (1Soltech ISTH-215-P) were used as the basis for the input and output datasets that were used in the design process for the MPPT technology. Both the standard irradiance (G_s) of the sun and the standard temperature (T_s) are assumed to be 1000 W/m^2 and 25 degrees Celsius, respectively. The irradiance levels that are regarded as the highest, or G_{\max} , and the lowest, or G_{\min} , are, respectively, 1000 W/m^2 and 0 W/m^2 . The highest temperature, denoted by the notation T_{\max} , is determined to be 35 degrees Celsius, while the lowest temperature, denoted by T_{\min} , is determined to be 15 degrees Celsius.

In Figure 6, the flowchart of proposed ANN algorithm is shown in order to implement the MPPT technology. In MATLAB/Simulink, the neural network toolbox function is used to construct the ANN algorithm. In order to collect data, develop and train a network, and assess the network performance, the fitting application of the ANN toolbox is utilized.

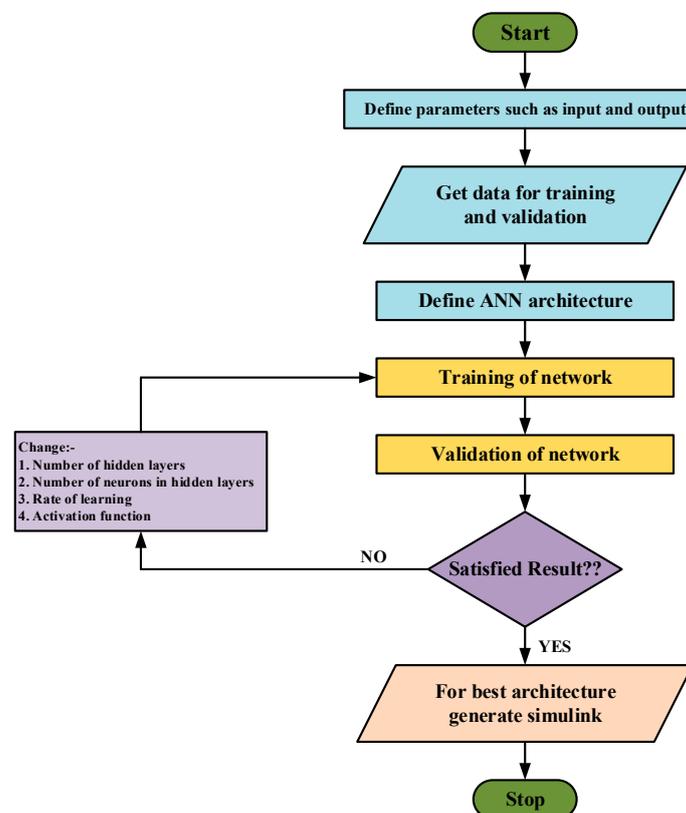


Figure 6. Flowchart of ANN Algorithm.

The created ANN is made up of sigmoid hidden neurons and linear output neurons, and it is built as a two-layer feedforward network. This kind of network is well-suited for solving multi-dimensional mapping issues. In order to train the neural network, one thousand datasets, including information on the temperature, irradiance and produced voltage of the chosen 1Soltech ISTH-215-P solar panel, are used. The data are then arbitrarily split into thirds: 10 percent for validation, 80 percent for training, and 10 percent for testing.

During the process of building the feedforward network, the number of neurons for the hidden layer is estimated to be 20. For training ANN datasets, a number of different training algorithms, including RP, GDM, LM, BR, SCG, and BFGS, may be used. The mean squared error, or MSE, is defined as the average squared difference between outputs and

objectives and is used by the LM algorithm, which is sometimes referred to as the damped least-squares approach.

In the LM strategy, “steepest descent” and “Gauss–Newton” are combined as two optimization techniques. When the estimated values are near to the final solution, the LM technique shifts to the Gauss–Newton method, which is initially more resistant to beginning values. This allows for a faster convergence rate. The transition from the steepest descent to the Gauss–Newton method is controlled by the “damping factor” parameter, ensuring the efficient use of the LM approach.

The parameters in question are adjusted at each iteration according to the Equation (8) [51]. The equation involves modifying the parameters based on the Jacobian matrix (J), the damping factor (λ), the identity matrix (I), the discrepancy between the network’s intended and actual performance (ϵ), the number of iterations (k), and the Hessian matrix ($J'J + \lambda_k I$).

$$\theta_{k+1} = \theta_k - \left(\frac{J' \epsilon}{J'J + \lambda_k I} \right)_{\theta=\theta_k} \quad (8)$$

The damping factor must be positive for the algorithm to converge. The LM algorithm is fast and has a built-in solution in MATLAB, making it efficient to use in that environment. The MSE is used to quantify the performance of the network, with a lower number signifying greater performance. The number 1 represents a perfect correlation, the value 0 represents a random association, and the regression coefficient R determines the degree to which the outputs and targets are correlated with one another.

Since they need fewer cross-validation steps, Bayesian regularized artificial neural networks (BRANNs) are a more reliable substitute for conventional backpropagation nets. BRANNs use mathematical methods to make ridge regression, such as nonlinear regression, a well-posed statistical problem. With this technique, only one iteration is needed to generate the model that is the “most generalizable”, but it must reach a local minimum instead of a global minimum. In contrast to the hundreds or thousands of repetitions required for unregularized ANNs, testing has shown that repeating the approach five times is sufficient to prevent any aberrant behavior. The mathematical model and all aspects of BRANN are described in detail in [52,53].

When it comes to training, the SCG strategy is one that is often used for a number of different kinds of issues. It employs knowledge of the second order rather than the line-search approach, which enables the amount of memory that is utilized to be decreased. Previous research [54,55], provides a full grasp of the approach by offering a detailed description of the final algorithm for the SCG. Similarly, when it comes to handle noisy and ill-conditioned data, resilient backpropagation (RP) is a robust optimization algorithm. RP uses only the sign of the gradient to update the weights and biases, which makes it computationally efficient and suitable for large-scale problems [56]. RP does not require any learning rate parameter to be tuned, which simplifies the training process and makes it less sensitive to hyperparameter tuning.

Broyden–Fletcher–Goldfarb–Shanno (BFGS) is a quasi-Newton optimization algorithm that approximates the inverse Hessian matrix of the loss function, which makes it converge faster than first-order optimization algorithms such as gradient descent [57]. Similar to the RP algorithm, BFGS does not require any learning rate parameter to be tuned, which simplifies the training process and makes it less sensitive to hyperparameter tuning. BFGS is relatively robust to noisy and ill-conditioned data and can handle non-convex optimization problems. Furthermore, for gradient descent momentum, the momentum parameter in GDM allows the algorithm to overcome local minima and converge faster to the global minimum of the loss function [58]. GDM reduces the oscillations and noise in the update direction and therefore provides a smoother convergence path. GDM is easy to implement and computationally efficient compared to other optimization algorithms.

4. Results and Discussion

The suggested model for solar energy collection involves simulating 1000 s of data transfer from a photovoltaic (PV) array to assess optimal analysis. This simulation uses a discrete approach instead of a continuous one. The accuracy of the artificial neural network (ANN) depends on the amount of training data and the selected training algorithm. Generally, larger training datasets result in less error from the ANN. The input data for the solar panel, consisting of solar irradiance and panel temperature, are supplied using a lookup table and a clock for synchronization.

The suitability of six algorithms for solar energy harvesting is also compared in this article. To assess the efficacy and performance of each method, regression, gradient, mean square error, μ , and validation check are all employed. Regression measures the output's ability to predict the inputs, and error is determined by deducting the goal from the output. The neural network uses three main types of samples: training, validation, and testing. Data are trained through training, and the network is then modified in response to error. By terminating training if faults are found, validation checks the network's generalizability. Contrarily, testing does not alter the training data and offers a neutral evaluation of the network's performance after training.

When referring to the training dataset, an "epoch" is a single cycle. A neural network has to be trained across a number of epochs. The iteration, or quantity of partitioned training data batches or steps required to complete one epoch, is connected to the epoch. Heuristically, the network has the opportunity to view the prior data and revise the parameters of the model. The model is impartial towards the most recent few data points during training. To adjust the ANN's parameters and keep the output divergence to a minimum, a gradient is a numerical computation. Each matrix or vector representation of the network parameters is employed as a reference point in this multivariable derivative of the loss function. The ANN sometimes encounters the local minimum issue and fails to converge; thus, the μ is added to the weight update phrase to prevent this issue. As a result, its value, which ranges from 0 to 1, directly influences the error of convergence during dataset training. The validation check is represented in training data as error minimization.

One or more selected error metrics are used to evaluate and validate a neural network (ANN) prediction model. An approximation of a function is achieved by the ANN algorithm via the use of a continuous error matrix, such as mean square error (MSE), mean absolute error (MAE), or root mean square error (RMSE).

After tallying up the mistakes across all of the inputs and outputs of the validation set, the results are then normalized based on the size of the set. Each data instance receives an application of a loss function that has been squared and then averaged across the entire dataset in order to maximize the predictive model's operation as a whole. By using error minimization, commonly known as "backpropagation," the ANN modifies its anticipated output in relation to its actual output.

4.1. Levenberg–Marquardt (LM)

Levenberg–Marquardt (LM) is an optimization algorithm widely used for training ANNs. It is particularly effective in solving non-linear regression problems and finding the optimal set of weights to minimize the difference between the network's predicted outputs and the target outputs. LM is known for its fast convergence and robustness, making it a popular choice in the field of machine learning. The LM algorithm combines the benefits of both the Gauss–Newton and steepest descent methods. It iteratively updates the weights of the network by considering both local and global search directions. The primary objective of LM is to minimize a given error function, often represented by the mean squared error (MSE), which quantifies the discrepancy between the predicted and target outputs. The core idea behind LM is to adaptively adjust the step size of weight updates based on the local curvature of the error surface. It achieves this by introducing a damping parameter that controls the trade-off between the local and global search directions. In regions where the error surface is steep and narrow, the damping parameter reduces the step size to

avoid overshooting the optimal solution. Conversely, in regions where the error surface is flat, the damping parameter increases the step size to speed up convergence. The LM algorithm starts with an initial set of weights and computes the Jacobian matrix, which represents the sensitivity of the network's outputs to changes in the weights. The Jacobian is used to approximate the Hessian matrix, which describes the curvature of the error surface. The Hessian matrix is modified by adding a damping term to ensure its positive definiteness, which guarantees the convergence of the algorithm. During each iteration, the LM algorithm updates the weights by solving a system of linear equations derived from the modified Hessian matrix and the gradient of the error function. This update step is performed iteratively until the error function reaches a minimum or a convergence criterion is met. The convergence criterion is often based on the change in the error function between iterations. One of the key advantages of LM is its ability to handle non-linear regression problems effectively. Unlike other gradient-based methods, the LM does not require explicit computation of the Hessian matrix, which can be computationally expensive for large networks. Instead, it approximates the Hessian using the Jacobian matrix and adapts the damping parameter to ensure stable convergence.

Another advantage of LM is its robustness to local minima. The combination of local and global search directions allows LM to escape from shallow local minima and find better solutions. This property makes LM particularly useful in situations where the error surface is complex and contains multiple local minima. Implementing LM for ANN training requires careful initialization of the weights and tuning of the damping parameter. In practice, the initial weights can be randomly assigned or set based on prior knowledge of the problem domain. The damping parameter is typically adjusted dynamically during the training process based on the convergence behavior of the algorithm. In conclusion, the Levenberg–Marquardt (LM) algorithm is a powerful optimization method for training artificial neural networks (ANNs). Its combination of local and global search directions, adaptive step size adjustment, and robustness to local minima make it an effective tool for solving non-linear regression problems. LM's fast convergence and stability contribute to its widespread use in the field of machine learning. With its ability to handle complex error surfaces and find optimal solutions, LM plays a significant role in the successful training of ANNs and the advancement of the field.

The plot in Figure 7 demonstrates the accuracy of the ANN's prediction of output in relation to input, indicated by the regression ($R = 1$) measurement. Error is defined as the variance between the solar panel's produced voltage output and its intended generated voltage, which is determined by deducting the output from the target. The regression plot illustrates that the LM algorithm has effectively trained the data with minimal error, as the output follows the desired value quite closely.

This approach for ANN is further validated by Figure 8, which shows zero error in the data matching training phase, validation phase, and test phases. The bins indicate how many vertical bars there are in the error histogram in Figure 8, where the total ANN error varies from -0.0004 (the leftmost bin) to 0.0000512 (the rightmost bin). It is shown how many samples from the chosen dataset fit into each of the 20 smaller bins that make up the error range. For 100 samples from the validation dataset, the bin with the error value -0.0000015 is in the center of the error histogram. The use of ANN for MPPT shows convergence at 20 bins with 0% error in the error histogram.

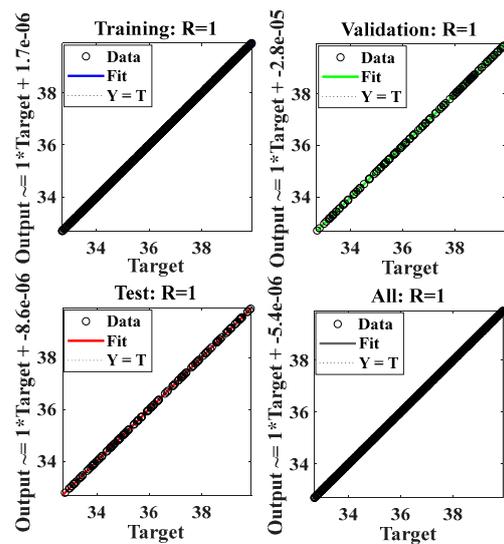


Figure 7. Regression plot of LM algorithm.

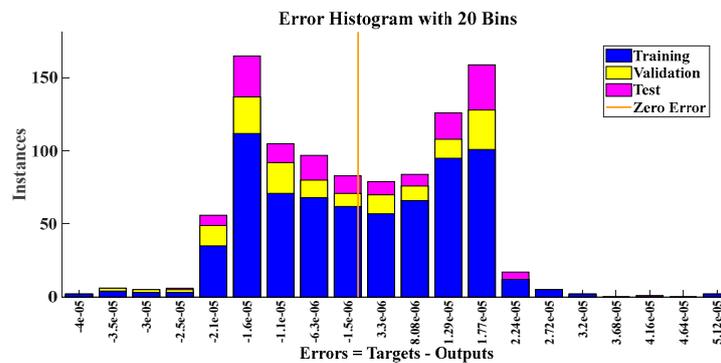


Figure 8. Error histogram plot of LM algorithm.

Figures 9 and 10 illustrate the training stage and performance phase of the ANN for the purpose of processing the selected dataset, respectively. The validation check, as well as the gradient and momentum parameter (Mu), for the training dataset are shown in Figure 9 at 1000 epochs. The simulation indicates that the gradient is 7.983×10^{-6} at the 1000 epoch, which represents the insignificant variances from the training data with a small loss function. The simulation’s findings indicate that a choice to produce zero outputs and the mean for each input vector make up the cumulative error. The LM algorithm suitability for MPPT is justified by the extremely low value of gradient, Mu and validation tests of the training dataset.

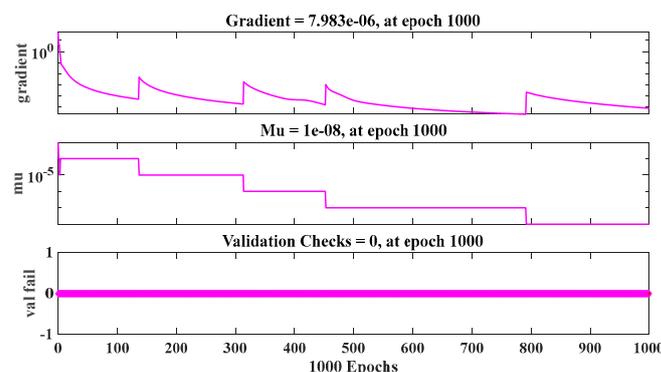


Figure 9. Training test plot of LM algorithm.

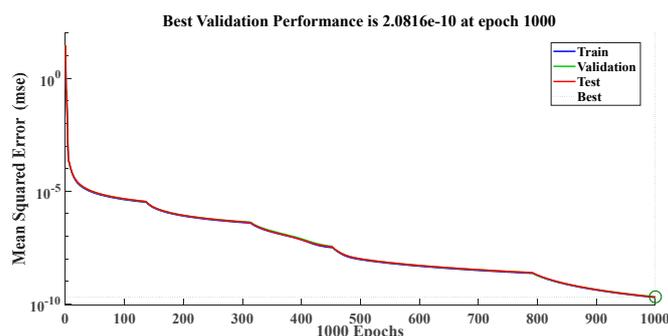


Figure 10. Performance plot of LM algorithm.

Figure 10 depicts the MSE for the trained dataset samples at different epochs, with the best training result being achieved at 1000 epochs. As a consequence of this, the trained dataset displays the highest level of validation performance after 1000 epochs have passed. In accordance with the findings of the simulation, the validation performance of 2.0816×10^{-10} , which is reached at the 1000 epoch, is the best possible value. The estimation of MPPT provided by the LM method had a nearly zero validation performance, which means that its error was minimal.

4.2. Bayesian Regularization (BR)

BR is a mathematical method like ridge regression that is used to solve nonlinear regression problems in a well-posed way. In this method, the nonlinear regression is turned into a statistical problem, and a conjugate gradient descent or a similar minimizer is used to find a solution. The advantage of using BR over traditional backpropagation methods is that it is more stable and eliminates the importance of thorough cross-validation. The method only requires a single iteration to create the model that is “most generalizable”, although repeating the process several times may be necessary to ensure a local minimum is reached instead of a global minimum. BRANN is a type of neural network that uses BR for training and is successful in solving a broad range of problems. It uses second-order knowledge instead of line-search methods, which uses less memory. The BR is a powerful technique used for training ANNs that addresses the challenges of overfitting and model complexity. By incorporating Bayesian principles into the training process, BR provides a probabilistic framework that allows for more robust and stable learning. The primary objective of BR is to find the optimal balance between fitting the training data well and avoiding overfitting. Overfitting occurs when the model becomes too complex and starts to memorize the training data instead of generalizing well to unseen data. BR tackles this issue by introducing a regularization term into the training objective, which encourages simpler and more robust models. The core idea behind BR is to impose a prior distribution over the weights of the network and update this distribution during the training process. The prior distribution reflects our prior beliefs about the values of the weights before observing the data. By incorporating prior knowledge, BR provides a regularization mechanism that constrains the model’s complexity and prevents it from overfitting.

During training, BR aims to find the posterior distribution of the weights given the observed data. This is achieved by maximizing the posterior probability using the training data and the prior distribution. The posterior distribution represents the updated beliefs about the weights after observing the data. The optimization process involves finding the weight values that maximize the posterior probability, which can be approached using various methods such as Markov chain Monte Carlo (MCMC) or variational inference. In conclusion, Bayesian regularization (BR) is a powerful technique for training artificial neural networks (ANNs) that addresses the challenges of overfitting and model complexity. By incorporating Bayesian principles, BR provides a probabilistic framework that allows for more robust and stable learning. Its ability to automatically determine the regularization strength and provide uncertainty estimates for predictions makes it a valuable tool in

the field of machine learning. With its adaptability and robustness, BR contributes to the advancement of ANN training and its successful application in various domains.

From the regression plot in Figure 11, the trained dataset does not need to go through a validation step. In the regression diagram, the best correlation between output and desired generated voltage is shown by the value $R = 1$, which indicates that the data for the solar PV system were appropriately trained.

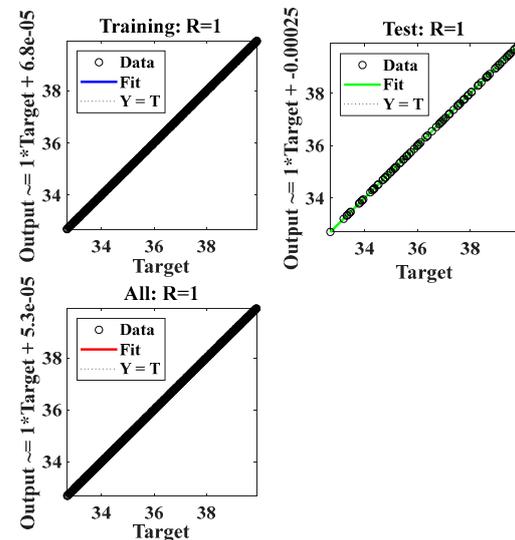


Figure 11. Regression plot of BR algorithm.

Figure 12 depicts the trained dataset, which contains zero errors in both the training and testing phases and total errors that range from -0.00111 (the bin on the left) to 0.00116 (the bin on the right) (the rightmost bin). The gradient and μ in the training state phase are, at the 1000th epoch, 0.00010493 and 5000 , respectively. The error histogram plot of the BR algorithm is shown in Figure 13; there are 20 smaller bins, and the central bin has a near-zero error of 0.000037 for 100 samples, which is larger than that of the LM approach.

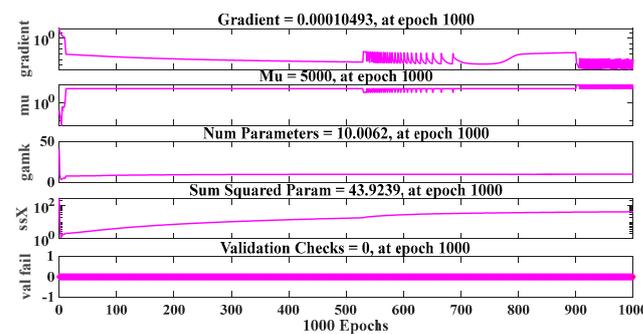


Figure 12. Training test plot of BR.

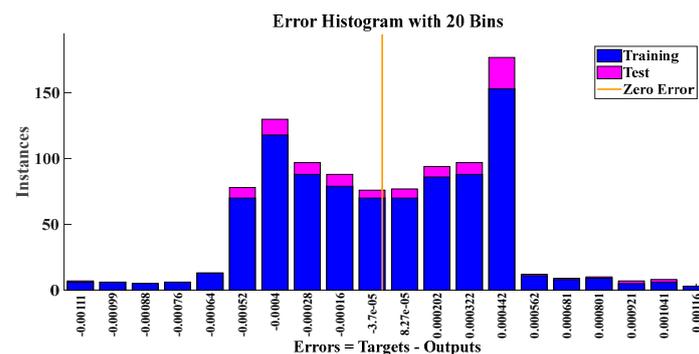


Figure 13. Error histogram plot of BR.

The actual number of parameters at the 1000th epoch is 10.0062; however, the total squared parameters at that time amount to 43.9239. This is shown by the LM method's slower backpropagation capacity for the training dataset and by the high value of Mu as well as an efficient number of parameters and absence of any validation tests.

Compared to the BR approach, which starts with an objective function for the prediction and adds the residual sum of squares and sum of squared weights for reducing the prediction error, the LM algorithm gives quicker convergence in predicting training data with near to zero error. As a consequence, the LM approach generally performs training dataset processing faster than the BR technique. Figure 14, which shows convergence of trained data with the optimum training outcome after 1000 iterations, shows the mean squared error at various epochs. The robustness of the BR method is shown by the best training performance of 1.582×10^{-7} at the 1000 epoch with no validation phase.

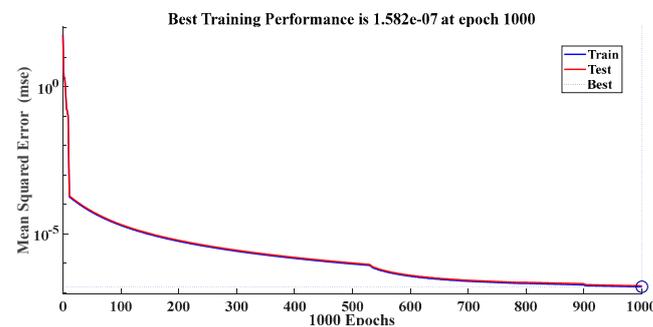


Figure 14. Performance plot of BR.

4.3. Scaled Conjugate Gradient (SCG)

The scaled conjugate gradient (SCG) algorithm is a powerful optimization technique commonly used for training artificial neural networks (ANNs). It offers an efficient and effective approach to updating the network's weights, facilitating convergence and improving the network's performance. ANNs are computational models inspired by the structure and functioning of the human brain. They are composed of interconnected nodes, known as neurons, which work together to process and transmit information. ANNs have gained significant attention in various fields due to their ability to learn from data and make accurate predictions. Training an ANN involves adjusting the weights of its connections to minimize the difference between the predicted outputs and the target outputs. The optimization algorithm used for weight adjustment plays a crucial role in determining the network's performance. The SCG algorithm is a popular choice for this task due to its desirable properties.

The SCG algorithm is based on the conjugate gradient (CG) method, a well-known optimization technique. The CG method aims to find the minimum of a function by iteratively updating the weight values. However, the SCG algorithm introduces a scaling factor that ensures the gradients have the same magnitude, which accelerates convergence. This scaling property allows the algorithm to adaptively adjust the learning rate for each weight update, leading to efficient weight adjustments and improved convergence speed. One of the key advantages of the SCG algorithm is that it eliminates the need for an explicit line search procedure, which is often required in traditional gradient-based optimization algorithms [59]. The line search procedure is responsible for determining an appropriate learning rate at each iteration. By incorporating second-order information and using a scaling approach, the SCG algorithm estimates the learning rate without the need for repetitive line search calculations. This feature significantly reduces the computational burden and makes the algorithm more efficient. The SCG algorithm exhibits excellent performance in handling non-linear optimization problems, making it particularly suitable for training ANNs with complex architectures. It has been successfully applied in various domains, including pattern recognition, data mining, and control systems. Researchers and practitioners often rely on the SCG algorithm to optimize the performance of their

neural network models and achieve accurate predictions in real-world applications. One notable advantage of the SCG algorithm is its robustness to noise and ill-conditioned problems. Ill-conditioned problems refer to situations where slight changes in input data or initial weights can have a significant impact on the optimization process. The SCG algorithm's adaptive learning rate and scaling properties help mitigate the effects of such issues, making it more stable and reliable. Another significant benefit of the SCG algorithm is its good generalization capability. Generalization refers to the ability of a trained network to perform well on unseen data. The SCG algorithm's efficient weight adjustments and convergence properties contribute to improved generalization, allowing the network to make accurate predictions on new and unseen instances. To summarize, the SCG algorithm offers several advantages for training ANNs. Its efficient weight adjustment scheme, adaptive learning rate estimation, and scaling properties contribute to faster convergence and improved generalization capabilities. The algorithm's robustness to noise and ill-conditioned problems makes it a reliable choice for various applications. Researchers and practitioners rely on the SCG algorithm to optimize the performance of their neural network models and achieve accurate predictions in diverse fields.

In comparison to the LM and BR algorithms, the R is somewhat less than 1, as seen by the regression plot in Figure 15. A regression value less than 1 for the SCG algorithm implies that there might be some level of inconsistency or error in the predictions made by the algorithm. The deviation from a perfect correlation suggests that there is room for improvement in the algorithm's ability to accurately predict the output based on the given input. In comparison, when the regression value is 1, it indicates a perfect correlation between the input and output data, suggesting that the algorithm is able to accurately capture the relationship between the variables.

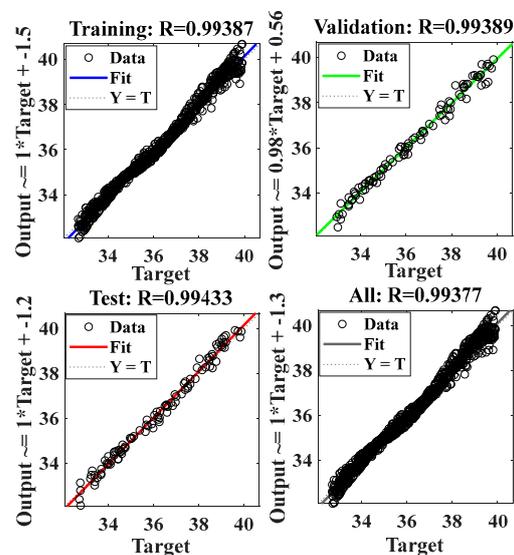


Figure 15. Regression plot of SCG algorithm.

As can be seen in Figure 16, the total error for the trained dataset varies from -0.7406 (the leftmost bin) to 0.8365 (the rightmost bin) when there is no mistake at any point throughout the training phase, validation phase, or test phase. The error histogram shows that the center bin has a value that is -0.006443 (100 samples) higher than the value produced by the LM and BR algorithms. After 75 iterations, Figure 17 displays a gradient with a value of 0.14161 , while the validation tests have a value of 6. When data training is terminated after 75 epochs, the SCG's performance in terms of achieving the target objective suffers as a direct result.

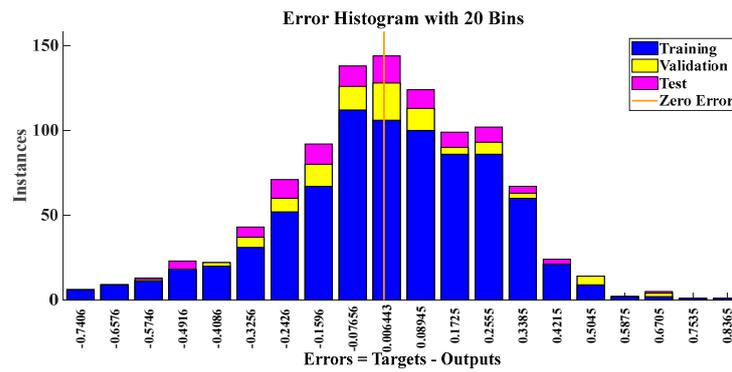


Figure 16. Error histogram of SCG algorithm.

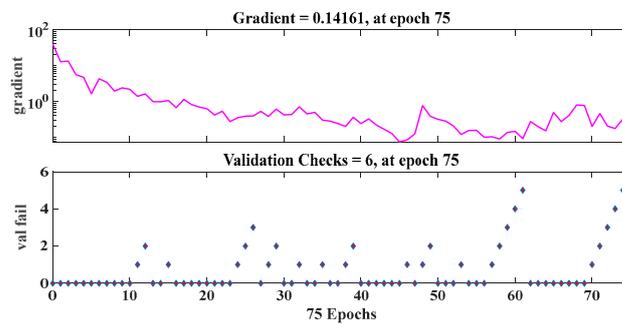


Figure 17. Training test plot of SCG.

The performance gradient for validation is higher than the maximum time for failure and lower than the lowest gradient. Figure 18, which represents the convergence of training data with the highest validation performance of 0.052985 at 75 epochs, displays the mean squared error at various epochs. The learned dataset for the solar PV system do not suit the SCG algorithm, despite having greater validation performance and a lower total training performance than the LM and BR algorithms. Similar to this, the BR method is less suitable than the LM algorithm due to the high processing time for prediction and the large momentum parameter.

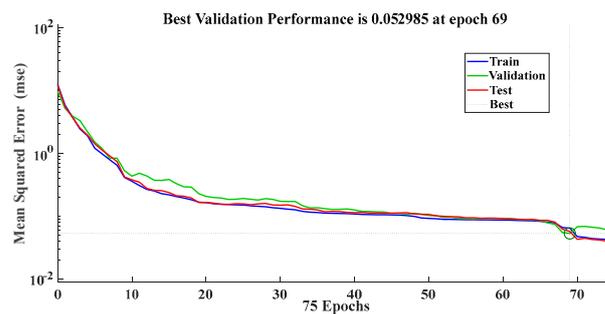


Figure 18. Performance plot of SCG.

4.4. Resilient Backpropagation (RP)

Resilient backpropagation (RP) is a gradient-based optimization algorithm commonly used in training artificial neural networks (ANNs). It is a variation of the traditional backpropagation algorithm that uses a different update rule for adjusting the weights of the neural network during training.

The key feature of the RP algorithm is its use of a dynamic learning rate for each weight in the network. The learning rate determines the step size used to update the weights during the backpropagation process and can have a significant impact on the convergence and stability of the training process. In traditional backpropagation, a fixed learning rate is

used for all weights, which can lead to slow convergence or even divergence if the learning rate is set too high or too low.

The RP algorithm overcomes this problem by using two different update rules based on the sign of the gradient of the error function with respect to the weight. If the gradient changes sign from one iteration to the next, indicating that the optimization process is moving in the wrong direction, the learning rate for that weight is reduced by a factor (e.g., 0.5). If the gradient has the same sign as the previous iteration, indicating that the optimization process is moving in the right direction, the learning rate is increased by a factor (e.g., 1.2). If the gradient is zero, indicating that the weight has reached a stationary point, the learning rate is unchanged.

This approach is known as a “resilient” update rule, as it allows the algorithm to recover quickly from bad updates and adapt to changing conditions during the optimization process. By adjusting the learning rate dynamically for each weight, the RP algorithm can achieve faster convergence and better stability compared to traditional backpropagation. Another advantage of the RP algorithm is its ability to handle noisy or ill-conditioned data, which can cause traditional optimization algorithms to become stuck in local minima. By adjusting the learning rate dynamically, the RP algorithm is able to navigate complex, high-dimensional search spaces more effectively and avoid becoming trapped in local minima. Hence, the RP algorithm is a robust and efficient optimization algorithm that is well-suited for training ANNs. Its ability to adapt to changing conditions and handle noisy data make it a popular choice for many machine learning applications.

From the regression plot in Figure 19, the trained dataset does not need to go through a validation step. In the regression diagram, the best correlation between output and desired generated voltage is shown by the value $R = 1$, which indicates that the data for the solar PV system were appropriately trained.

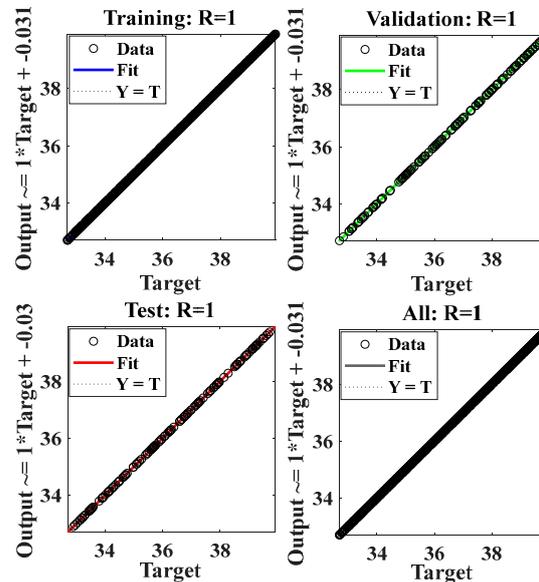


Figure 19. Regression plot of RP algorithm.

After 21 iterations, Figure 20 displays a gradient with a value of 0.033415, while the validation tests have a value of 6 at 21 epochs. A gradient value of 0.033415 indicates the rate of change or steepness of the objective function with respect to the algorithm’s parameters. A low gradient value indicates that the algorithm is making progress towards the optimal solution, and the estimate is close to the true minimum. This value suggests that the RP algorithm is making significant progress towards optimizing the objective function and reaching the desired solution. The validation tests having a value of 6 at 21 epochs implies that the algorithm’s predictions during the validation phase have a relatively larger deviation from the true values compared to desired accuracy. This indicates that the

algorithm may not be performing optimally in terms of accurately predicting the target values during the validation phase.

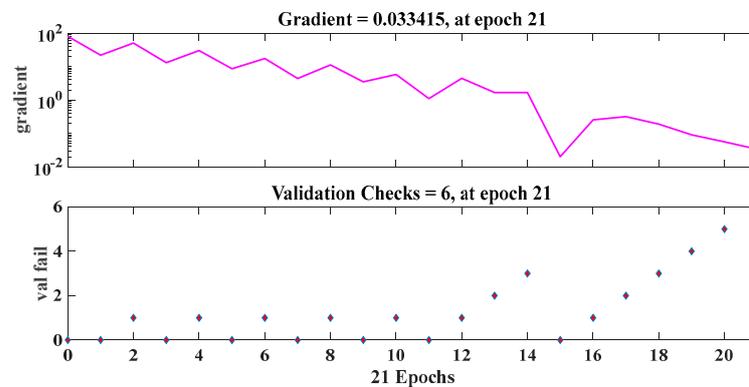


Figure 20. Training plot of RP algorithm.

Figure 21 displays the mean squared error at a number of different epochs. This figure illustrates the convergence of trained data with the optimal training result after 15 iterations. The greatest training performance of 2.8668×10^{-6} was achieved at the 15 epoch even though there was no validation phase, which demonstrates the resilience of the RP algorithm.

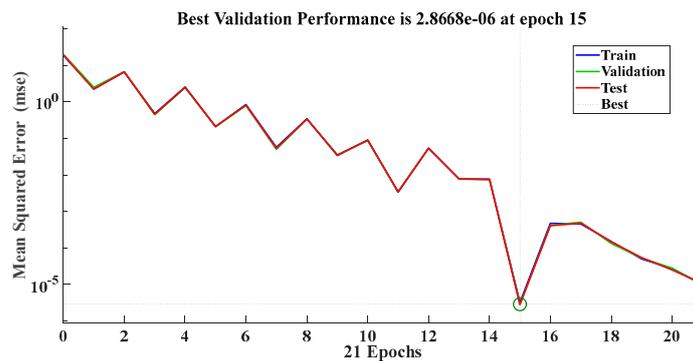


Figure 21. Performance test plot of Rp algorithm.

4.5. Gradient Descent Momentum (GDM)

Gradient descent momentum (GDM) is a popular optimization algorithm for training artificial neural networks (ANNs). It is a variation of the traditional gradient descent algorithm that incorporates a momentum term to help speed up the convergence of the optimization process. The momentum term allows the algorithm to “remember” the direction it has been moving in the past and use that information to help accelerate the optimization process.

The basic idea behind gradient descent with momentum is to add a “velocity” term to the weight updates that takes into account the direction and magnitude of previous weight updates. Specifically, the velocity for weight ‘ w ’ at iteration ‘ t ’ is given by:

$$V_t(w) = \alpha \times V_{t-1}(w) - \eta \times \nabla \in (w_t) \quad (9)$$

where α is the momentum coefficient (typically set to a value between 0 and 1), η is the learning rate, $\in (w_t)$ is the error function at iteration t , and $\nabla \in (w_t)$ is the gradient of the error function with respect to weight w at iteration t .

The weight update at iteration t is then given by:

$$w_{t+1} = w_t + V_t(w) \quad (10)$$

The momentum term allows the algorithm to “smooth out” the weight updates over time, reducing the impact of small, noisy changes in the gradient and helping to avoid getting stuck in local minima. The momentum term can also help the algorithm to accelerate when the gradient is pointing consistently in the same direction, allowing it to move more quickly towards the global minimum. One of the key advantages of GDM is its ability to handle noisy or ill-conditioned data, which can cause traditional optimization algorithms to become stuck in local minima. By incorporating information about the direction and magnitude of previous weight updates, the momentum term allows the algorithm to navigate complex, high-dimensional search spaces more effectively and avoid becoming trapped in local minima. Another advantage of gradient descent with momentum is its ability to handle non-convex objective functions. Non-convex objective functions can be difficult to optimize using traditional gradient descent, as the optimization process can become stuck in local minima. By incorporating information about the direction and magnitude of previous weight updates, the momentum term allows the algorithm to explore more of the search space and avoid becoming trapped in local minima. Hence, gradient descent with momentum is a powerful and efficient optimization algorithm that is well-suited for training ANNs. Its ability to handle noisy or ill-conditioned data and to navigate complex, high-dimensional search spaces make it a popular choice for many machine learning applications.

In comparison to the LM, RP, BFGS, and BR algorithms, the R is somewhat less than 1, as seen by the regression plot in Figure 22 for GDM algorithm.

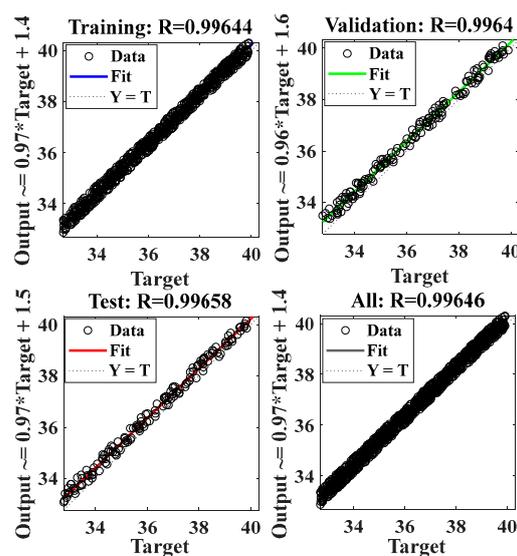


Figure 22. Regression plot of GDM algorithm.

A regression value of 0.996 indicates a strong positive correlation between the predicted values and the actual values in the training dataset, while a regression value of 1 indicates a perfect positive correlation between the predicted values and the actual values.

Figure 23 displays a gradient with a value of 11.485, while the validation tests have a value of 6 at 8 epochs. The higher value of gradient for GDM algorithm indicates that the current estimate is far from the true minimum, and the algorithm needs to take large steps to get closer to it. This could result in overshooting the minimum or becoming stuck in a local minimum instead of the global minimum, which can lead to poor performance.

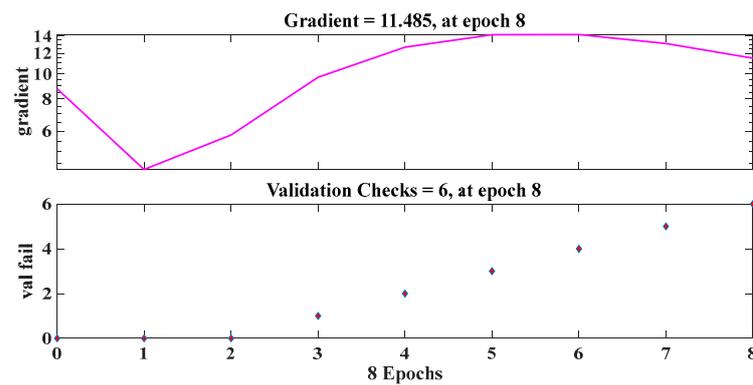


Figure 23. Training plot of GDM algorithm.

Figure 24 displays the performance plot of GDM algorithm. The plot shows that the algorithm achieves its best validation performance of 0.15378 at 2 epochs. A validation performance of 0.15378 indicates that the algorithm's predictions have a relatively moderate level of deviation from the true values of the validation dataset. Although not as low as achieving near-zero validation performance, this value still demonstrates a reasonable level of accuracy in estimating the desired output within a very short training time.

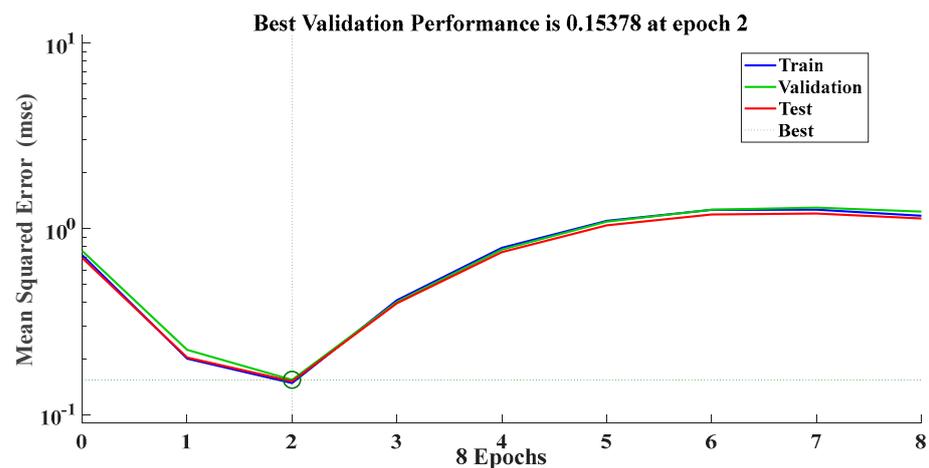


Figure 24. Performance plot of GDM algorithm.

4.6. Broyden–Fletcher–Goldfarb–Shanno (BFGS) Quasi Newton

The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is a popular quasi-Newton method for unconstrained optimization, commonly used in training artificial neural networks (ANNs). The basic idea behind the BFGS algorithm is to approximate the inverse Hessian matrix of the objective function, which is a measure of how the gradient of the function changes as the weights of the network are updated. By using this approximation, the algorithm is able to estimate the direction and magnitude of weight updates that are likely to lead to a decrease in the objective function.

The BFGS algorithm works by iteratively updating an approximation of the inverse Hessian matrix, denoted by H_k , at each iteration k . The weight update at iteration k is then given by:

$$W_{k+1} = W_k - \alpha_k \times H_k \times \nabla \in (W_k) \quad (11)$$

where α_k is a step size parameter, $\nabla \in (W_k)$ is the gradient of the objective function with respect to the weights at iteration k , and H_k is the approximate inverse Hessian matrix.

The BFGS algorithm uses a rank-2 update formula to update the approximate inverse Hessian matrix at each iteration. Specifically, the update formula is given by:

$$H_{k+1} = (1 - P_k \times S_k \times Y_k^T) \times H_k (1 - P_k \times Y_k \times S_k^T) + P_k \times S_k \times S_k^T \quad (12)$$

where $S_k = W_{k+1} - W_k$, $Y_k = \nabla \in (W_{k+1}) - \nabla \in (W_k)$, and $P_k = \frac{1}{(Y_{kT} \times S_k)}$. This update formula is designed to approximate the true inverse Hessian matrix, while also maintaining the positive definiteness of the approximation.

One of the key advantages of the BFGS algorithm is its ability to handle non-convex objective functions, which are commonly encountered in ANNs. Non-convex objective functions can be difficult to optimize using traditional gradient descent methods, as the optimization process can become stuck in local minima. The BFGS algorithm is able to overcome this problem by using an approximation of the inverse Hessian matrix, which allows it to navigate complex, high-dimensional search spaces more effectively.

Another advantage of the BFGS algorithm is its ability to handle ill-conditioned or noisy data. Traditional optimization algorithms can be sensitive to noisy or ill-conditioned data, which can cause them to become stuck in local minima. The BFGS algorithm is able to overcome this problem by using an approximation of the inverse Hessian matrix, which allows it to smooth out noisy or ill-conditioned data and navigate the search space more effectively. Hence, the BFGS algorithm is a powerful and efficient optimization algorithm that is well-suited for training ANNs. Its ability to handle non-convex objective functions and ill-conditioned or noisy data makes it a popular choice for many machine learning applications. However, like all optimization algorithms, it may require careful tuning of the step size parameter and other hyperparameters to achieve optimal performance on a given problem.

Figure 25 is a figure that indicates how accurate the ANN's prediction of output in relation to input is. This accuracy is demonstrated by the regression measurement of $R = 1$, which is shown in the plot. The regression plot demonstrates that the BFGS method has successfully trained the data with a small amount of error by displaying the output in a manner that closely reflects the intended value. In other words, the BFGS algorithm accurately captures the relationship between the input variables (such as solar irradiance and temperature) and the corresponding output variable (such as voltage and power output in energy harvesting applications). This high regression value signifies that the BFGS algorithm is effectively modeling and predicting the behavior of the energy harvesting system, making it a reliable and precise algorithm for optimizing energy extraction from solar photovoltaic systems.

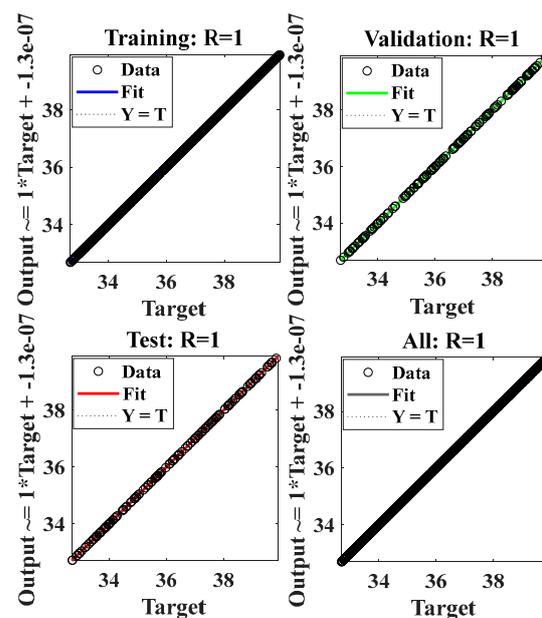


Figure 25. Regression plot of BFGS algorithm.

In Figure 26, a training plot of the BFGS algorithm is shown. The validation check as well as the gradient and reset values for the training dataset are shown at 10 epochs.

The algorithm attained a gradient with a value of 1.0211×10^{-7} , while the validation tests and resets had no value at 10 epochs. A gradient value of 1.0211×10^{-7} indicates that the algorithm's estimate of the optimal solution is very close to the true minimum. In general, a lower gradient value suggests that the algorithm is converging well and making small, steady steps towards the optimal solution. This indicates that the algorithm is effectively adjusting its parameters to minimize the difference between predicted and actual values, resulting in improved accuracy. The fact that the validation tests and resets have no value at 10 epochs suggests that the algorithm has successfully learned and generalized the underlying patterns in the training data. This means that the algorithm's predictions align well with the validation dataset, and there is no need for further adjustments or resets at this point. Overall, these findings indicate that the algorithm has achieved a high level of accuracy and convergence, making it a reliable choice for MPPT (maximum-power point tracking) energy harvesting in solar PV systems.

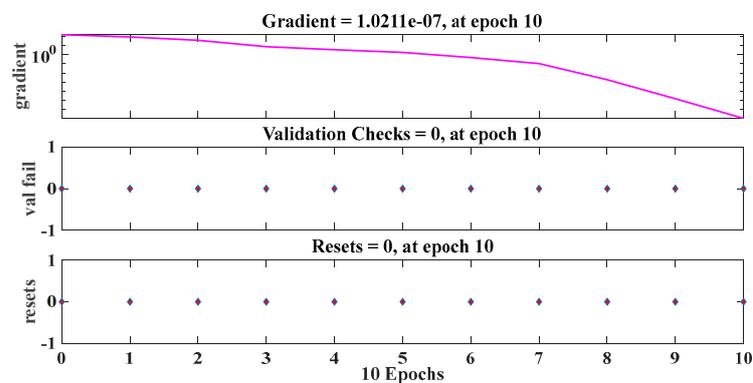


Figure 26. Training plot of BFGS algorithm.

In Figure 27, The convergence of training data with the highest performance value of the BFGS algorithm is displayed. The algorithm attained the best validation performance of 9.98×10^{-17} at 10 epochs. A validation performance of 9.98×10^{-17} indicates that the algorithm's predictions are extremely close to the true values of the validation dataset. This near-zero validation performance suggests that the algorithm has successfully learned the underlying patterns and relationships in the training data, and it can generalize its predictions accurately to unseen data. It also suggests that the algorithm is capable of accurately capturing the complex relationships between input variables and the desired output, enabling it to make precise predictions for MPPT (maximum-power point tracking) energy harvesting in solar PV systems.

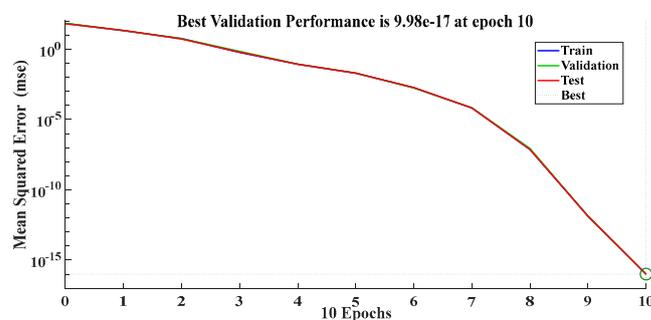


Figure 27. Performance plot of BFGS algorithm.

In relation to solar radiation and array temperature, respectively, Figure 28a–d show the output power and load power of the PV array. According to the MPPT topology of ANN, both powers are claimed to follow the solar irradiation and the temperature of the PV array. Based on the amount of simulation time that has passed, the PV array's

produced power, load power, irradiance, and temperature are shown here. The MPPT is satisfied when the array temperature is 15 degrees Celsius and the sun irradiance is 200 watts per square meter. This results in the lowest power output of 100 watts. With an array temperature of 35 degrees Celsius and a solar irradiation of 1000 watts per square meter, the power output reaches its maximum of 450 watts. The ripples in the output power are not smoothed out by this model since it does not have filter component. As a direct consequence of this, the power waveforms that are created and loaded have discernible ripples.

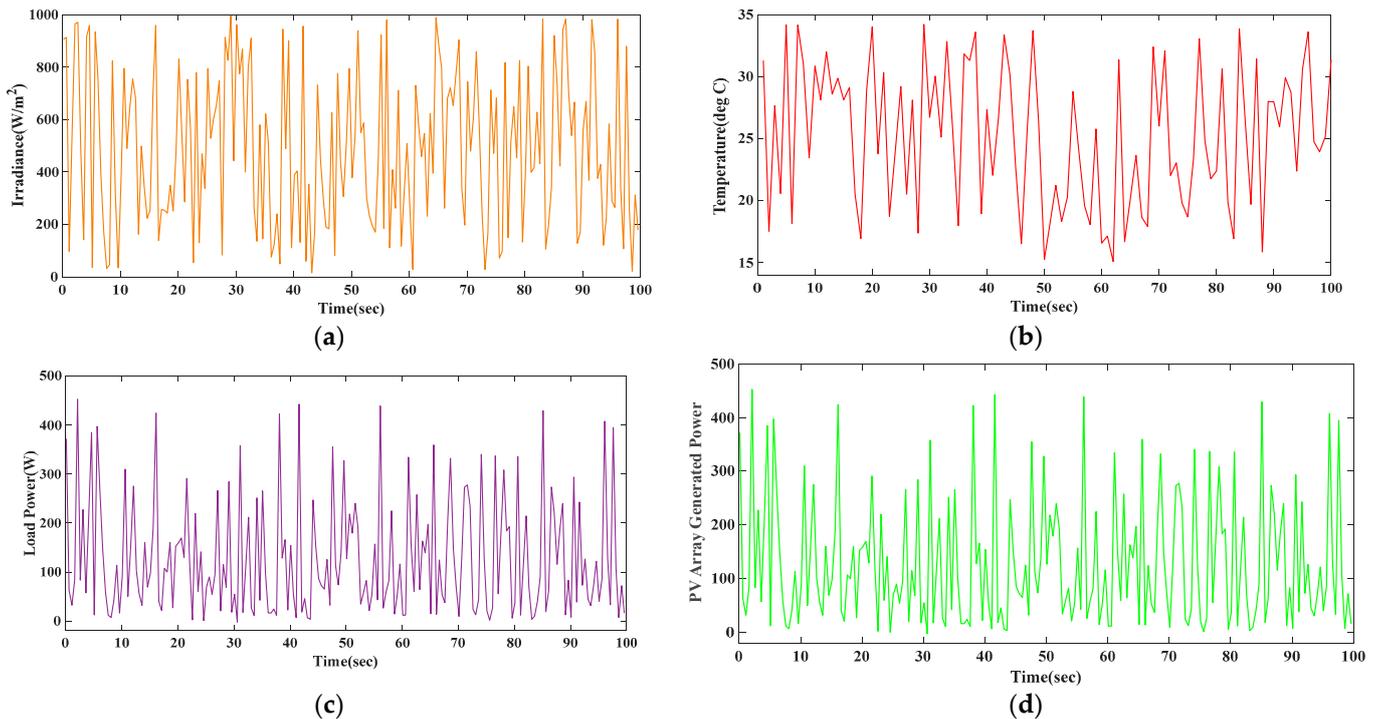


Figure 28. (a) Irradiance (W/m^2) versus time (s) plot of the solar panel, (b) temperature ($^{\circ}C$) versus time (s) plot of the solar panel, (c) load power (W) versus time (s) plot of the solar panel, (d) and power generated (W) versus time (s) of the panel.

Table 2 presents a dataset consisting of matching values for solar irradiance (G) in watts per square meter (W/m^2), temperature (T) in degrees Celsius ($^{\circ}C$), and maximum voltage (V_{max}) in volts (V). The table contains 120 rows, each representing a specific observation or data point. These 120 data points are sample data among 1000 data points. Using Equations (3)–(5) of solar irradiance, temperature, and maximum voltage, the data in Table 2 were generated. These equations are used to calculate the solar irradiance (G), temperature (T), and maximum voltage (V_{MP}) based on specific parameters and random values. Furthermore, the data were used to train the neural network.

Table 2. ANN databases of matching values of solar irradiance, G (W/m^2), temperature, T ($^{\circ}C$), and maximum voltage, V_{max} (V).

S. No.	Solar Irradiance (W/m^2)	Temperature ($^{\circ}C$)	Maximum Voltage (V_{max})
1	905.7919371	31.29447373	34.02775793
2	913.3758561	17.53973633	38.99308058
3	97.540405	27.64718492	35.34439271
4	546.8815192	20.56996438	37.89919856

Table 2. Cont.

S. No.	Solar Irradiance (W/m ²)	Temperature (°C)	Maximum Voltage (V _{max})
5	964.8885352	34.15013671	32.99689215
6	970.5927818	18.15226163	38.77196507
7	485.3756487	34.14333896	32.99934607
8	141.8863386	31.00560938	34.13203507
9	915.7355252	23.43522565	36.86486789
10	959.4924264	30.84414659	34.19032152
11	35.71167857	28.11481398	35.1755833
12	933.9932478	31.98258612	33.77935624
13	757.7401306	28.5747031	35.00956793
14	392.2270195	29.86264936	34.54463221
15	171.1866878	28.1095578	35.17748073
16	31.83284638	29.12092176	34.81238845
17	46.17139063	20.5384597	37.91057143
18	823.4578283	16.94263562	39.20862797
19	317.0994801	28.89657246	34.89337631
20	34.4460805	34.00444098	33.04948685
21	381.5584571	23.77488719	36.74225347
22	795.1999011	30.31033576	34.38302189
23	489.7643958	18.73745209	38.56071717
24	646.3130101	23.91172401	36.69285675
25	754.686682	29.18729662	34.78842779
26	679.7026769	20.52050154	37.91705415
27	162.6117352	28.10196008	35.18022343
28	498.364052	17.37995363	39.05076054
29	340.3857267	34.19487917	32.98074057
30	223.8119395	26.70535502	35.68438389
31	255.0951155	30.02534119	34.48590209
32	699.0767227	25.11914103	36.25699128
33	959.2914252	32.81806505	33.4777567
34	138.6244428	25.9443106	35.95911332
35	257.5082541	17.98588011	38.83202714
36	254.282179	31.81434512	33.84008956
37	243.5249687	31.28569652	34.03092641
38	349.983766	33.58527246	33.20080249
39	251.083858	18.93190501	38.49052161
40	473.2888489	27.32089352	35.46218065
41	830.8286279	22.03319014	37.37098869
42	549.7236083	26.70528182	35.68441031
43	285.8390188	33.34387328	33.28794519
44	753.7290943	30.14400458	34.44306579

Table 2. Cont.

S. No.	Solar Irradiance (W/m ²)	Temperature (°C)	Maximum Voltage (V _{max})
45	567.8216407	22.60891694	37.16315707
46	53.95011867	16.51708579	39.3622472
47	779.1672301	25.61595106	36.07764783
48	129.9062085	33.68021368	33.16652966
49	469.3906411	26.37647322	35.80310693
50	337.1226444	15.23804139	39.82396944
51	794.2845407	18.24364616	38.73897617
52	528.5331355	21.22430084	37.66298964
53	601.9819414	18.31297459	38.7139493
54	654.0790985	20.25942569	38.01129992
55	748.1515928	28.78429006	34.93390913
56	83.821378	24.01083197	36.65707977
57	913.3373615	19.57953937	38.25673208
58	825.8169775	18.04756038	38.80976118
59	996.1347166	25.76684871	36.02317529
60	442.6782698	16.56351058	39.34548832
61	961.8980809	17.1330554	39.13988833
62	774.9104647	15.09268448	39.87644183
63	868.6947054	31.34606441	34.00913421
64	399.7826491	16.68871691	39.30029008
65	800.0684802	20.19740806	38.03368767
66	910.6475944	23.62827655	36.79517845
67	263.8029165	18.63694057	38.59700083
68	136.0685587	17.91077961	38.85913767
69	579.7045874	32.38584415	33.63378412
70	144.9547982	25.99720404	35.94001931
71	622.0551315	32.06062235	33.75118594
72	513.2495399	22.01904762	37.376094
73	75.96669169	23.03616068	37.00892636
74	123.3189348	19.79832307	38.17775335
75	239.9525257	18.67815577	38.58212255
76	49.65443033	23.34534138	36.89731521
77	944.7871897	33.0543222	33.39247023
78	489.2526384	24.81728185	36.36595943
79	900.0538464	21.7543882	37.4716334
80	111.2027553	22.38493562	37.24401209
81	389.738837	30.60504137	34.27663612
82	403.9121456	19.83382572	38.16493725
83	131.9732926	16.9290905	39.21351762
84	956.1345402	33.84101182	33.10848314

Table 2. Cont.

S. No.	Solar Irradiance (W/m ²)	Temperature (°C)	Maximum Voltage (V _{max})
85	59.77954295	26.5041719	35.75700899
86	353.1585712	19.69559827	38.21483598
87	15.40343765	31.4238808	33.98104327
88	168.9900295	15.86047603	39.59927676
89	731.7223857	27.9823095	35.22341609
90	450.9237064	27.95491926	35.2333037
91	296.3208056	25.94017785	35.9606052
92	188.955015	29.89385614	34.53336687
93	183.5111557	28.73550867	34.95151873
94	625.6185607	22.36969193	37.24951491
95	81.12576887	30.6045487	34.27681396
96	775.7126786	33.58771942	33.19991917
97	435.8585886	24.73583265	36.39536177
98	306.349472	23.93567499	36.68421069
99	510.7715642	25.17017311	36.23856921
100	794.8314169	31.35255417	34.00679147
101	378.6093827	27.8863626	35.25805196
102	532.8255888	31.23160917	34.05045141
103	939.001562	22.01454207	37.37772046
104	550.1563429	32.51885623	33.58576809
105	587.0447045	27.44950172	35.41575437
106	301.2463303	19.15484585	38.41004219
107	230.4881602	24.41846697	36.50992761
108	194.7642896	31.88617585	33.81415938
109	170.7080471	19.51843562	38.27878993
110	435.6986841	19.55328596	38.2662093
111	923.3796421	21.22204573	37.66380371
112	184.8163201	23.60414783	36.80388868
113	979.7483784	33.09761937	33.37684038
114	111.1192234	23.77739946	36.74134657
115	408.7198461	20.16129392	38.04672451
116	262.2117478	26.89792148	35.61486932
117	711.2157804	27.05686179	35.55749346
118	117.4176509	19.43493468	38.30893293
119	318.7783019	20.93351746	37.76795953
120	507.8582847	23.48333519	36.84750083

Table 3 compares the efficacy of six different ANN algorithms—the LM algorithm, BR algorithm, SCG algorithm, RP algorithm, GDM algorithm, and BFGS algorithm—in this regard.

Table 3. Comparison of the ANN algorithms' performance.

Parameters	Algorithm					
	LM	RP	SCG	BR	GDM	BFGS
Regression	1	1	0.993	1	0.996	1
Error at middle bin	−0.0000015	-	−0.006443	−0.000037	-	-
Gradient	7.983×10^{-6}	0.033415	0.14161	0.00010493	11.485	1.0211×10^{-7}
Performance	2.0816×10^{-10}	2.8668×10^{-6}	0.052985	1.583×10^{-7}	0.15378	9.98×10^{-17}
Momentum parameter	1.00×10^{-8}	-	-	5000	-	-
Epochs	1000	15	69	1000	2	10

The table shows various performance parameters for each algorithm, including regression, error at the middle bin, gradient, performance, momentum parameter, and epochs. The regression value for the LM, BR, RP, and BFGS algorithms is 1, indicating a perfect correlation between the input and output data, while the SCG and GDM algorithms have a slightly lower regression value of 0.993 and 0.996, respectively. The error at the middle bin is negative for the LM algorithm (−0.0000015), indicating that the algorithm is performing better than the other algorithms. The gradient value for the BFGS algorithm (1.0211×10^{-7}) is much lower than that for the BR, RP, SCG, LM, and GDM algorithms, which are 0.00010493, 0.033415, 0.14161, 7.983×10^{-6} , and 11.485, respectively, which indicates that the BFGS algorithm has better convergence properties. The lower gradient value of BFGS indicates that the algorithm is making progress towards the optimal solution, and the estimate is close to the true minimum. This suggests that the algorithm is converging well and making small, steady steps towards the optimal solution. Similarly, the higher gradient value of the GDM algorithm indicates that the current estimate is far from the true minimum, and the algorithm needs to take large steps to get closer to it. This could result in overshooting the minimum or becoming stuck at a local minimum instead of the global minimum, which can lead to poor performance.

The performance value for the BFGS algorithm (9.98×10^{-17}) is significantly better than that for the BR algorithm (1.583×10^{-7}), SCG algorithm (0.052985), RP algorithm (2.8668×10^{-6}), GDM algorithm (0.15378), and LM algorithm (2.0816×10^{-10}), indicating that the BFGS algorithm provides the most accurate predictions. The LM algorithm secures second place in terms of performance and accuracy of predictions. The momentum parameter is only specified for the LM and BR algorithms, with the LM algorithm having a value of 0.00000001 and the BR algorithm having a value of 5000. The momentum parameter was not applicable for RP, SCG, GDM, and BFGS algorithms, as they do not use momentum. Finally, looking at the number of epochs required for convergence, the RP and BR algorithms required the fewest epochs at 15 and 2, respectively. The LM, GDM, and BFGS algorithms required moderate numbers of epochs, while the SCG algorithm required a relatively large number of epochs at 69. Overall, the results illustrate that the LM algorithm outperforms the other two algorithms in terms of accuracy, convergence, and performance, making it the best choice for MPPT energy harvesting in a solar PV system.

Overall, the results suggest that the LM and BFGS algorithms are the best performers in terms of gradient and performance, while the RP and BR algorithms also perform well across the various metrics. The SCG and GDM algorithms appear to be less effective for this particular problem.

5. Conclusions

In conclusion, this paper presented a novel approach for evaluating the efficacy of six ANN algorithms (LM, BR, RP, GDM, BFGS, and SCG) in the context of MPPT for solar PV systems. The authors utilized a two-layer feedforward neural network from the ANN toolbox and trained it with generated data on solar irradiance, panel temperature, and generated voltage. Based on the outcomes, the regression values of the LM, RP, BR, and BFGS algorithms are all equal to 1, whereas the regression values for the SCG and GDM

algorithms are less than 1. The gradient values for the LM, RP, BFGS, SCG, BR, and GDM algorithms are 7.983×10^{-6} , 0.033415, 1.0211×10^{-7} , 0.14161, 0.00010493, and 11.485, respectively. Likewise, the performance values for these algorithms are 2.0816×10^{-10} , 2.8668×10^{-6} , 9.98×10^{-17} , 0.052985, 1.583×10^{-7} , and 0.15378. Overall, the findings demonstrated that the LM and BFGS algorithms performed exceptionally well in terms of performance and gradient, with LM showing almost no error in the middle epoch. At 1000 epochs, the validation efficiency was highest, indicating close-to-zero values for the momentum parameter, validation, and gradient checks. The LM and BFGS algorithms exhibit superior performance in terms of regression, gradient, and overall performance. Additionally, the RP and BR algorithms also demonstrate good performance across various metrics, while the SCG and GDM algorithms show relatively less effectiveness in addressing the proposed problem. These findings provide valuable insights into the relative performance of the six evaluated algorithms for MPPT energy harvesting in solar PV systems.

The proposed ANN-based MPPT energy harvesting concept holds potential for solving problems on a large scale and can be incorporated into multilayer neural networks, making it highly versatile. Furthermore, the study revealed a perfect correlation between input and output data, exemplified by the consistency between generated power and load power after MPPT. This ANN-based MPPT energy harvesting model can find applications in standalone and grid-connected solar PV systems, as well as military equipment, telecommunications, and space satellites. The suggested ANN-based MPPT energy harvesting model exhibits applicability across various domains and can be integrated into numerous technologies. Additionally, the model can be utilized for solar radiation and temperature prediction, energy estimation, energy management in smart homes and cities, and forecasting solar radiation and temperature. Looking ahead, the future scope of solar PV MPPT based on ANN appears promising, offering several areas for research and development. One potential direction is exploring the use of reinforcement learning algorithms to further optimize MPPT control in solar PV systems. Additionally, integrating AI-based control systems with power electronics devices, such as DC–DC converters and inverters, could enhance overall system efficiency. Advancements in sensors and data acquisition systems hold the potential to develop more sophisticated and accurate ANN models for MPPT control. Furthermore, investigating hybrid control algorithms that combine multiple AI-based techniques could lead to improved performance in solar PV MPPT. The proposed work offers valuable insights but has limitations. Firstly, it focuses only on six specific ANN algorithms, potentially limiting the understanding of the overall algorithmic landscape. Secondly, the evaluation is based on specific performance parameters, which may not capture all aspects of algorithm performance. Additionally, the study focuses solely on solar PV systems, excluding other renewable energy sources. This limits the applicability of the findings to a broader range of energy harvesting systems. Overall, this research contributes valuable insights into the evaluation of ANN algorithm performance for MPPT in solar PV systems and sets the stage for future advancements in the field.

Author Contributions: Conceptualization, A.S. and M.T.; formal analysis, M.T.H., A.S., M.T., S.U. and M.A.H.; funding acquisition, S.U. and A.B.; investigation, M.T.H., A.S., M.T. and S.U.; methodology, M.T.H., A.S., M.T. and M.A.H.; project administration, M.T., S.U. and A.B.; resources, M.T.; supervision, A.S. and M.T.; validation, M.T.H., A.S., M.T., A.B. and M.A.H.; writing—original draft, M.T.H., A.S. and M.T.; writing—review and editing, S.U., A.B. and M.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University (43-PRFA-P-38), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research project was funded by the Deanship of Scientific Research, Princess Nourah bint Abdulrahman University, through the Program of Research Project Funding After Publication, grant No (43-PRFA-P-38).

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

References

1. Kouro, S.; Leon, J.I.; Vinnikov, D.; Franquelo, L.G. Grid-Connected Photovoltaic Systems: An Overview of Recent Research and Emerging PV Converter Technology. *IEEE Ind. Electron. Mag.* **2015**, *9*, 47–61. [[CrossRef](#)]
2. Tarroja, B.; Shaffer, B.; Samuelsen, S. The importance of grid integration for achievable greenhouse gas emissions reductions from alternative vehicle technologies. *Energy* **2015**, *87*, 504–519. [[CrossRef](#)]
3. Rokonuzzaman; Mishu, M.K.; Amin, N.; Nadarajah, M.; Roy, R.B.; Rahman, K.S.; Buhari, A.M.; Binzaid, S.; Shakeri, M.; Pasupuleti, J. Self-Sustained Autonomous Wireless Sensor Network with Integrated Solar Photovoltaic System for Internet of Smart Home-Building (IoSHB) Applications. *Micromachines* **2021**, *12*, 653. [[CrossRef](#)]
4. Koochi-Kamali, S.; Rahim, N.; Mokhlis, H.; Tyagi, V. Photovoltaic electricity generator dynamic modeling methods for smart grid applications: A review. *Renew. Sustain. Energy Rev.* **2016**, *57*, 131–172. [[CrossRef](#)]
5. De Giorgi, M.G.; Congedo, P.M.; Malvoni, M. Photovoltaic power forecasting using statistical methods: Impact of weather data. *IET Sci. Meas. Technol.* **2014**, *8*, 90–97. [[CrossRef](#)]
6. Karami, N.; Moubayed, N.; Outbib, R. General review and classification of different MPPT Techniques. *Renew. Sustain. Energy Rev.* **2017**, *68*, 1–18. [[CrossRef](#)]
7. Mishu, M.K.; Rokonuzzaman, M.; Pasupuleti, J.; Shakeri, M.; Rahman, K.S.; Hamid, F.A.; Tiong, S.K.; Amin, N. Prospective Efficient Ambient Energy Harvesting Sources for IoT-Equipped Sensor Applications. *Electronics* **2020**, *9*, 1345. [[CrossRef](#)]
8. Rokonuzzaman; Shakeri, M.; Hamid, F.A.; Mishu, M.K.; Pasupuleti, J.; Rahman, K.S.; Tiong, S.K.; Amin, N. IoT-Enabled High Efficiency Smart Solar Charge Controller with Maximum Power Point Tracking—Design, Hardware Implementation and Performance Testing. *Electronics* **2020**, *9*, 1267. [[CrossRef](#)]
9. Sivakumar, P.; Kader, A.A.; Kaliavaradhan, Y.; Arutchelvi, M. Analysis and enhancement of PV efficiency with incremental conductance MPPT technique under non-linear loading conditions. *Renew. Energy* **2015**, *81*, 543–550. [[CrossRef](#)]
10. Roy, R.B.; Basher, E.; Yasmin, R.; Rokonuzzaman, M. Fuzzy logic based MPPT approach in a grid connected photovoltaic system. In Proceedings of the the 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), Dhaka, Bangladesh, 18–20 December 2014; pp. 1–6. [[CrossRef](#)]
11. Bhatti, A.R.; Salam, Z.; Sultana, B.; Rasheed, N.; Awan, A.B.; Sultana, U.; Younas, M. Optimized sizing of photovoltaic grid-connected electric vehicle charging system using particle swarm optimization. *Int. J. Energy Res.* **2018**, *43*, 500–522. [[CrossRef](#)]
12. Dixit, T.V.; Yadav, A.; Gupta, S. Experimental assessment of maximum power extraction from solar panel with different converter topologies. *Int. Trans. Electr. Energy Syst.* **2018**, *29*, e2712. [[CrossRef](#)]
13. Rokonuzzaman, H.-E.-H. Design and implementation of maximum power point tracking solar charge controller. In Proceedings of the 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 22–24 September 2016; pp. 1–5. [[CrossRef](#)]
14. Mishu, M.K.; Rokonuzzaman; Pasupuleti, J.; Shakeri, M.; Rahman, K.S.; Binzaid, S.; Tiong, S.K.; Amin, N. An Adaptive TE-PV Hybrid Energy Harvesting System for Self-Powered IoT Sensor Applications. *Sensors* **2021**, *21*, 2604. [[CrossRef](#)] [[PubMed](#)]
15. Esram, T.; Chapman, P.L. Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques. *IEEE Trans. Energy Convers.* **2007**, *22*, 439–449. [[CrossRef](#)]
16. Basha, C.H.; Rani, C. Different Conventional and Soft Computing MPPT Techniques for Solar PV Systems with High Step-Up Boost Converters: A Comprehensive Analysis. *Energies* **2020**, *13*, 371. [[CrossRef](#)]
17. Lopez-Guede, J.M.; Ramos-Hernanz, J.; Altun, N.; Ozdemir, S.; Kurt, E.; Azkune, G. Neural Modeling of Fuzzy Controllers for Maximum Power Point Tracking in Photovoltaic Energy Systems. *J. Electron. Mater.* **2018**, *47*, 4519–4532. [[CrossRef](#)]
18. Khanam, J.J.F.; Simon, Y. Modeling of a photovoltaic array in MATLAB simulink and maximum power point tracking using neural network. *J. Electr. Electron. Syst.* **2018**, *7*, 40–46.
19. Chen, L.; Wang, X. Enhanced MPPT method based on ANN-assisted sequential Monte-Carlo and quickest change detection. *IET Smart Grid* **2019**, *2*, 635–644. [[CrossRef](#)]
20. Chtouki, I.; Wira, P.; Zazi, M. ICIT—Comparison of several neural network perturb and observe MPPT methods for photovoltaic applications. In Proceedings of the 2018 IEEE International Conference on Industrial Technology (ICIT), Lyon, France, 20–22 February 2018; pp. 909–914. [[CrossRef](#)]
21. Bouakkaz, M.S.; Boukadoum, A.; Boudebouz, O.; Bouraiou, A.; Attoui, I. ANN based MPPT Algorithm Design using Real Operating Climatic Condition. In Proceedings of the 2020 2nd International Conference on Mathematics and Information Technology (ICMIT), Adrar, Algeria, 18–19 February 2020. [[CrossRef](#)]
22. Algarín, C.R.; Hernández, D.S.; Leal, D.R. A Low-Cost Maximum Power Point Tracking System Based on Neural Network Inverse Model Controller. *Electronics* **2018**, *7*, 4. [[CrossRef](#)]

23. Divyasharon, R.; Banu, R.N.; Devaraj, D. Artificial Neural Network based MPPT with CUK Converter Topology for PV Systems Under Varying Climatic Conditions. In Proceedings of the 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 11–13 April 2019; pp. 1–6. [\[CrossRef\]](#)
24. Fatima, K.; Alam, M.A.; Minai, A.F. Optimization of Solar Energy Using ANN Techniques. In Proceedings of the 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), Greater Noida, India, 18–19 October 2019. [\[CrossRef\]](#)
25. Al-Majidi, S.D.; Abbod, M.F.; Al-Raweshidy, H.S. Design of an intelligent MPPT based on ANN using a real photovoltaic system data. In Proceedings of the 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, 3–6 September 2019. [\[CrossRef\]](#)
26. Elgendy, M.A.; Atkinson, D.J.; Zahawi, B. Experimental investigation of the incremental conductance maximum power point tracking algorithm at high perturbation rates. *IET Renew. Power Gener.* **2016**, *10*, 133–139. [\[CrossRef\]](#)
27. Jyothy, L.P.; Sindhu, M.R. An Artificial Neural Network based MPPT Algorithm For Solar PV System. In Proceedings of the 2018 4th International Conference on Electrical Energy Systems (ICEES), Chennai, India, 7–9 February 2018. [\[CrossRef\]](#)
28. Vimalarani, C.; Kamaraj, N. Improved method of maximum power point tracking of photovoltaic (PV) array using hybrid intelligent controller. *Optik* **2018**, *168*, 403–415. [\[CrossRef\]](#)
29. Roy, R.B.; Cros, J.; Nandi, A.; Ahmed, T. Maximum Power Tracking by Neural Network. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020. [\[CrossRef\]](#)
30. Danyali, S.; Aghaei, O.; Shirkhani, M.; Aazami, R.; Tavooosi, J.; Mohammadzadeh, A.; Mosavi, A. A New Model Predictive Control Method for Buck-Boost Inverter-Based Photovoltaic Systems. *Sustainability* **2022**, *14*, 11731. [\[CrossRef\]](#)
31. Aazami, R.; Heydari, O.; Tavooosi, J.; Shirkhani, M.; Mohammadzadeh, A.; Mosavi, A. Optimal Control of an Energy-Storage System in a Microgrid for Reducing Wind-Power Fluctuations. *Sustainability* **2022**, *14*, 6183. [\[CrossRef\]](#)
32. Kaya, C.B.; Kaya, E.; Gökkuş, G. Training Neuro-Fuzzy by Using Meta-Heuristic Algorithms for MPPT. *Comput. Syst. Sci. Eng.* **2023**, *45*, 69–84. [\[CrossRef\]](#)
33. Ibnelouad, A.; El Kari, A.; Ayad, H.; Mjahed, M. Improved cooperative artificial neural network-particle swarm optimization approach for solar photovoltaic systems using maximum power point tracking. *Int. Trans. Electr. Energy Syst.* **2020**, *30*, 12439. [\[CrossRef\]](#)
34. Roy, R.B.; Rokonuzzaman, M.; Amin, N.; Mishu, M.K.; Alahakoon, S.; Rahman, S.; Pasupuleti, J. A Comparative Performance Analysis of ANN Algorithms for MPPT Energy Harvesting in Solar PV System. *IEEE Access* **2021**, *9*, 102137–102152. [\[CrossRef\]](#)
35. Li, Z.; Rahman, S.M.; Vega, R.; Dong, B. A Hierarchical Approach Using Machine Learning Methods in Solar Photovoltaic Energy Production Forecasting. *Energies* **2016**, *9*, 55. [\[CrossRef\]](#)
36. Xia, L.; Ma, Z.; Kokogiannakis, G.; Wang, Z.; Wang, S. A model-based design optimization strategy for ground source heat pump systems with integrated photovoltaic thermal collectors. *Appl. Energy* **2018**, *214*, 178–190. [\[CrossRef\]](#)
37. Ding, M.; Wang, L.; Bi, R. An ANN-based Approach for Forecasting the Power Output of Photovoltaic System. *Procedia Environ. Sci.* **2011**, *11*, 1308–1315. [\[CrossRef\]](#)
38. Porrizzo, R.; Cipollina, A.; Galluzzo, M.; Micale, G. A neural network-based optimizing control system for a seawater-desalination solar-powered membrane distillation unit. *Comput. Chem. Eng.* **2013**, *54*, 79–96. [\[CrossRef\]](#)
39. Chine, W.; Mellit, A.; Lughfi, V.; Malek, A.; Sulligoi, G.; Pavan, A.M. A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. *Renew. Energy* **2016**, *90*, 501–512. [\[CrossRef\]](#)
40. Vaz, A.; Elsinga, B.; van Sark, W.; Brito, M. An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands. *Renew. Energy* **2016**, *85*, 631–641. [\[CrossRef\]](#)
41. Elsheikh, A.H.; Sharshir, S.W.; Abd Elaziz, M.; Kabeel, A.E.; Guilan, W.; Haiou, Z. Modeling of solar energy systems using artificial neural network: A comprehensive review. *Sol. Energy* **2019**, *180*, 622–639. [\[CrossRef\]](#)
42. Subudhi, B.; Pradhan, R. A Comparative Study on Maximum Power Point Tracking Techniques for Photovoltaic Power Systems. *IEEE Trans. Sustain. Energy* **2013**, *4*, 89–98. [\[CrossRef\]](#)
43. De Brito, M.A.G.; Galotto, L.; Sampaio, L.P.; Melo, G.D.A.E.; Canesin, C.A. Evaluation of the Main MPPT Techniques for Photovoltaic Applications. *IEEE Trans. Ind. Electron.* **2013**, *60*, 1156–1167. [\[CrossRef\]](#)
44. Kollimalla, S.K.; Mishra, M.K. A Novel Adaptive P&O MPPT Algorithm Considering Sudden Changes in the Irradiance. *IEEE Trans. Energy Convers.* **2014**, *29*, 602–610. [\[CrossRef\]](#)
45. Elgendy, M.A.; Zahawi, B.; Atkinson, D.J. Assessment of the Incremental Conductance Maximum Power Point Tracking Algorithm. *IEEE Trans. Sustain. Energy* **2013**, *4*, 108–117. [\[CrossRef\]](#)
46. Elobaid, L.M.; Abdelsalam, A.K.; Zakzouk, E.E. Artificial neural network-based photovoltaic maximum power point tracking techniques: A survey. *IET Renew. Power Gener.* **2015**, *9*, 1043–1063. [\[CrossRef\]](#)
47. Mohammed, K.K.; Buyamin, S.; Shams, I.; Mekhilef, S. Maximum power point tracking based on adaptive neuro-fuzzy inference systems for a photovoltaic system with fast varying load conditions. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12904. [\[CrossRef\]](#)
48. Dragicevic, T.; Wheeler, P.; Blaabjerg, F. Artificial Intelligence Aided Automated Design for Reliability of Power Electronic Systems. *IEEE Trans. Power Electron.* **2019**, *34*, 7161–7171. [\[CrossRef\]](#)

49. Ramana, V.V.; Jena, D. Maximum power point tracking of PV array under non-uniform irradiance using artificial neural network. In Proceedings of the 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Kozhikode, India, 19–21 February 2015; pp. 1–5. [[CrossRef](#)]
50. Ali, F.; Sarwar, A.; Bakhsh, F.I.; Ahmad, S.; Shah, A.A.; Ahmed, H. Parameter extraction of photovoltaic models using atomic orbital search algorithm on a decent basis for novel accurate RMSE calculation. *Energy Convers. Manag.* **2023**, *277*, 116613. [[CrossRef](#)]
51. Dkhichi, F.; Oukarfi, B.; Fakkar, A.; Belbounaguia, N. Parameter identification of solar cell model using Levenberg–Marquardt algorithm combined with simulated annealing. *Sol. Energy* **2014**, *110*, 781–788. [[CrossRef](#)]
52. Gimazov, R.; Shidlovskiy, S. Simulation Modeling of Intelligent Control Algorithms for Constructing Autonomous Power Supply Systems with Improved Energy Efficiency. In *MATEC Web of Conferences*; EDP Sciences: Les Ulis, France, 2018; Volume 155, p. 01032. [[CrossRef](#)]
53. Macaulay, J.; Zhou, Z. A Fuzzy Logical-Based Variable Step Size P&O MPPT Algorithm for Photovoltaic System. *Energies* **2018**, *11*, 1340. [[CrossRef](#)]
54. Cho, T.-H.; Hwang, H.-R.; Lee, J.-H.; Lee, I.-S. Comparison of Intelligent Methods of SOC Estimation for Battery of Photovoltaic System. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 90907. [[CrossRef](#)]
55. Ul-Haq, A.; Sindi, H.F.; Gul, S.; Jalal, M. Modeling and Fault Categorization in Thin-Film and Crystalline PV Arrays Through Multilayer Neural Network Algorithm. *IEEE Access* **2020**, *8*, 102235–102255. [[CrossRef](#)]
56. Abd Allah, A.R.; Essai, M.H.; Yahya, A. Comparison of different backpropagation training algorithms using robust M-estimators performance functions. In Proceedings of the 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 23–24 December 2015; pp. 384–388. [[CrossRef](#)]
57. Dong, C.L.; Nocedal, J. On the limited memory BFGS method for large scale optimization. *Math. Program* **1989**, *45*, 503–528. Available online: <https://link.springer.com/content/pdf/10.1007%2FBF01589116.pdf> (accessed on 11 June 2023).
58. Luo, X.; Qin, W.; Dong, A.; Sedraoui, K.; Zhou, M. Efficient and High-quality Recommendations via Momentum-incorporated Parallel Stochastic Gradient Descent-Based Learning. *IEEE/CAA J. Autom. Sin.* **2020**, *8*, 402–411. [[CrossRef](#)]
59. Møller, M.F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* **1993**, *6*, 525–533. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.