IJRCS

International Journal of Robotics and Control Systems
Vol. 2, No. 3, 2022, pp. 562-580
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Short-Term Solar PV Power Generation Day-Ahead Forecasting Using Artificial Neural Network: Assessment and Validation

Abdel-Nasser Sharkawy [a,1,*], Mustafa M. Ali [a,2], Hossam H. H. Mousa [b,3], Ahmed S. Ali [c,4], G. T. Abdel-Jaber [d,5]

[a] Mechatronics Engineering, Department of Mechanical Engineering, South Valley University, Qena 83523, Egypt
[b] Department of Electrical Engineering, South Valley University, Qena 83523, Egypt
[c] Mechatronics Engineering, Department of Mechanical Engineering, Assiut University, Assiut, Egypt
[d] Department of Mechanical Engineering, South Valley University, Qena 83523, Egypt
[1] abdelnassersharkawy@eng.svu.edu.eg; [2] M.Ali@eng.svu.edu.eg; [3] H.herzallah@eng.svu.edu.eg;
[4] ahmadsaad01@yahoo.com; [5] gtag2000@yahoo.com
* Corresponding Author

ARTICLE INFO

Article history
Received July 16, 2022
Revised August 27, 2022
Accepted September 03, 2022

Keywords
Power Prediction;
Multilayer Feedforward NN;
Solar PV Power Station;
Levenberg-Marquardt Algorithm;
Error Backpropagation Algorithm;
MLFFNN Effectiveness

ABSTRACT

Solar photovoltaics (PV) is considered an auspicious key to dealing with energy catastrophes and ecological contamination. This type of renewable energy is based on climatic conditions to produce electrical power. In this article, a multilayer feedforward neural network (MLFFNN) is implemented to predict and forecast the output power for a solar PV power station. The MLFFNN is designed using the module temperature and the solar radiation as the two main only inputs, whereas the expected power is its output. Data of approximately one week (6-days) are obtained from a real PV power station in Egypt. The data of the first five days are used to train the MLFFNN. The training of the designed MLFFNN is executed using two types of learning algorithms: Levenberg-Marquardt (LM) and error backpropagation (EBP). The data of the sixth day, which are not used for the training, are used to check the efficiency and the generalization capability of the trained MLFFNN by both algorithms. The results provide evidence that the trained MLFFNN is running very well and efficiently to predict the power correctly. The results obtained from the trained MLFFNN by LM (MLFFNN-LM) are compared with the corresponding ones obtained by the MLFFNN trained by EBP (MLFFNN-EBP). From this comparison, the MLFFNN-LM has slightly lower performance in the training stage and slightly better performance in the stage of effectiveness investigation compared with the MLFFNN-EBP. Finally, a comparison with other previously published approaches is presented. Indeed, predicting the power correctly using the artificial NN is useful to avoid the fall of the power that maybe happen at any time.

Highlights

- The multilayer feed-forward neural network is proposed for short-term solar PV energy projects.
- The MLFFNN training is carried out using two algorithms and confirmed using real data from a solar PV plant in Egypt.
- The lowest mean squared error (MSE) and training error are attained.

- The better performance and effectiveness of the proposed model are proven for solar power forecasting.

Nomenclature

Meaning	Abbreviation	Meaning	Abbreviation
Error Back Propagation	EBP	Neural Network	NN
Feed-Forward NN	FFNN	Photovoltaic	PV
Generalized regression NN	GRNN	Renewable energy sources	RESs
Levenberg-Marquardt	LM	Support Vector Machine	SVM
Mean Squared Error	MSE	Support Vector Regression	SVR
Mean Absolute Error	MAE	Multi-Layer FFNN	MLFFNN
Root Mean Squared Error	RMSE	Training Error	TE

1. Introduction

Due to the urgent global energy crisis and increased environmental pollution levels, eradicating the usage of fossil fuels and expanding the utilization of green energy are critical issues. So, the Paris Agreement, which came into force in November 2016, has exhibited a robust strategy for limiting global warming issues by forcing industrial countries to wipe out traditional energy sources (TESs) and hang on to a circular economy. Hence, present and future research trends are applied to combine renewable energy sources (RESs) into the utility grid and to reduce the TESs hazards [1].

According to the continuous variations of the environmental conditions, various types of RESs' control schemes are implemented to cope with the increased electrical power demand [2], [3]. By using wind power generation, it is essential to accurately specify the wind direction and speed to precipitously extract all available power. At the same time, the thermal solar and Photovoltaic (PV) systems, solar irradiation, air temperature, and humidity are the most important factors that control the power generation and the PV system's performance. Here, the stability of the electrical power system's performance is the major challenge because of the increased penetration of RESs into modern power grids, especially solar PV systems. Forecasting the output power from various RESs is the main obstacle to power grid stability by developing an advanced control methodology.

Corresponding to solar PV systems expansion, as presented in Fig. 1, it is required to deal with the stochastic nature of solar irradiation, temperature, and surrounding environmental conditions [5]. The continuous solar power variables fluctuations have several significant influences on the power networks in terms of operation, control, and planning [4].

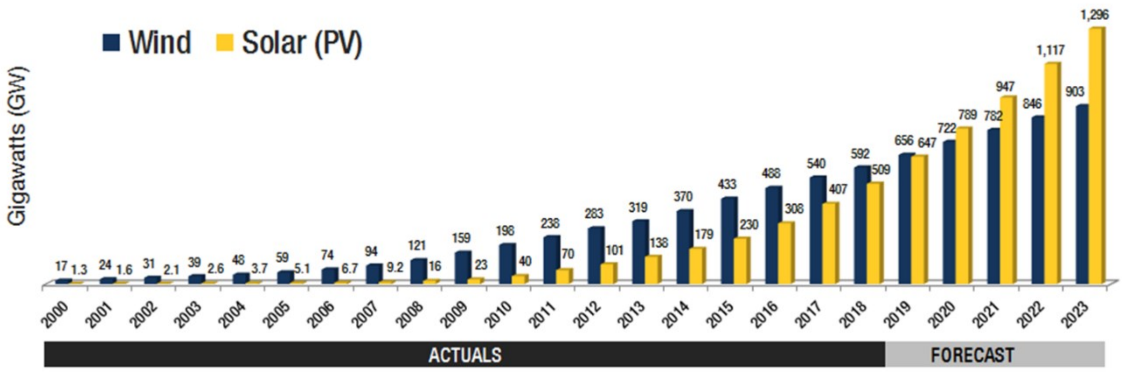


Fig. 1. Solar PV and wind energy annual capacity [4]. The drawing is carried out using the online website: http://www.fi-powerweb.com/Renewable-Energy.html

Several researchers have investigated two choices for controlling the performance indices of utility grid while integrating solar PV output power: firstly, installing huge storage systems (such as batteries, pump-hydro, and so on), whereas second is to develop accurate models for energy production forecasting based on the climate conditions. Therefore, accurate forecasting of PV-generated power, whether directly or indirectly, is a vital challenge for grid system stability, reliability, and optimization instead of utilizing new energy storage components which are financial benefits are questionable. Looking specifically at the climate changes, which have serious impacts on the power output, it is necessary to predict these continuous changes. Hence, the classification of the prediction horizon through the PV output power based on time is presented in Fig. 2. The prediction horizon can be defined as short-term, medium-term, or long-term based on the judgment-creating actions in the intelligent grid, which may be changeable based on the designers' assumptions and usages [5]. The operational region and forecast horizon are the most important factors in deciding on a strategy for the prediction of PV power.

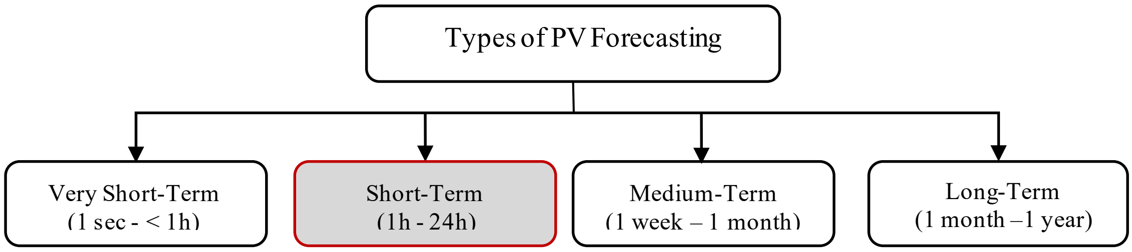


Fig. 2. Types of PV power prediction based on time [6], [7].

Energy planners and researchers employed a variety of methodologies and techniques to estimate and predict the PV output power. These methods were implemented based on mathematical analysis, one diode model using four parameters, a partial functional linear regression model, and finally, machine learning such as support vector machine (SVM) and neural network (NN), as discussed in Ref. [6].

Mathematical prediction methods in [8]–[11], such as the persistence model and statistical approaches, were investigated. The persistence model is based on historical data. In statistical approaches, time series models, regression methods, and regression trees are employed. However, these methods provided poor predicting accuracy. Furthermore, this validation of non-linear data is a very complex task. Another mathematical approach is implemented based on the information of the solar PV systems, for instance, operational pre-stored data, location, and variations of the climate variables [12], [13]. This approach has better accuracy in case of the climate conditions are steady. On the other hand, this prediction model is highly sensitive to weather variations. In addition, these models must be intended especially for a particular PV system and location. In [14], KUMAR et al. forecasted the amount of power produced by the PV system based on one diode model using four parameters using solar radiation and the module temperature. The results from this model showed that the error between the predicted power and the measured one is high at lower radiation, whereas it is low at a high one. Guochang Wang et al. [15] employed a regularized partial functional linear regression model for predicting a one-day-ahead solar PV generation system. In this approach, the knowledge about the climatical agents, such as the mean atmospheric pressure and insolation, is valuable in the power prediction of the solar PV system. Their results proved that the power prediction significantly declined the prediction error. However, considering period-changing pressure and insolation as efficient interpreters were missing.

Recently, methods based on machine learning, for instance, SVM and NNs, have been suggested for predicting solar PV power. SVM is considered a supervised machine-learning technique depending on the theory of structural risk minimization [16]. So, it is widely used for classification and regression tasks in which it is nominated as Support Vector Regression (SVR). In [17], Yen et al. predicted one-hour ahead of solar PV output power based on SVM with random forest. Their approaches are based on ecological constraints, for example, temperature, humidity, rainfall, and wind speed. The results showed that the random forest-based approach had better prediction accuracy than the SVM.

NN is also used for predicting the solar PV output power, which has the properties that it can approximate any function as well as its ability of generalization under different conditions [18], [19]. Kumar et al. [20] developed three NNs (Elman NN, feed-forward (FF)NN, and Generalized regression (GR)NN) to forecast the power of grid-connected semi-transparent solar PV system using several inputs, namely, solar cell location, solar radiation, the wind velocity, and the ambient temperature. Their acquired results are stated that the NN produces accurate prediction with a root-mean-square error of 0.25 in ELMAN NN and 0.30 in FFNN, and 0.426 in GRNN. The effectiveness and the generalization ability of these NNs are not investigated and evaluated under different conditions. NNs also is proposed for power prediction in [21]–[23]. The main gap in these studies is that the NN assessment under nature conditions variation is not confirmed and investigated.

From the above discussion, it is concluded that the prediction of solar PV power production using artificial NNs, and deep learning algorithms requires more deep investigation and analysis. The training or approximation errors should be small and close to the value of zero. Therefore, the prediction's accuracy is improved and increased. In addition, the size of the input layer should be small. In other meaning, the algorithm should have few inputs. This can minimize the complexity and the calculations. The generalization ability and the effectiveness of the prediction method should be investigated and verified under different conditions and cases than the training case.

The main contribution and novelty of this article can be summarized as follows:

- The correct prediction for the solar PV output power can be used to avoid power outages at any time because of environmental conditions and interruptions.
- A simple multi-layer (ML)FFNN is proposed and designed to predict the output solar PV power using only two main parameters (the module temperature and the solar radiation).
- The MLFFNN's training is confirmed using real data from a solar PV plant in Egypt. The training process is carried out using data from five days obtained from a real solar PV power station in Egypt. Hence, the main criterion for the best training is obtaining the lowest mean-squared error (MSE) and the lowest training error (TE), which are close to the zero value.
- Both Levenberg-Marquardt (LM) and error-back-propagation (EBP) algorithms are used and investigated for training the designed MLFFNN and compared.
- The generalization capability and the effectiveness of the trained MLFFNN-LM and MLFFNN-EBP are then checked and investigated using different data than the ones used for the training process. The results show that both MLFFNN-LM and MLFFNN-EBP are trained very well, and the MSE and the training error are very low. Therefore, the trained MLFFNN can predict the power in an accurate way under any condition.
- A comparative study is presented between the obtained results by MLFFNN-LM and MLFFNN-EBP. Furthermore, another comparison is presented between them and other previously published methods.

The rest of the article is divided as follows: Section 2 gives a mathematical analysis for calculating the output power of the solar PV plant. In Section 3, the MLFFNN structure is studied. In Section 4 and Section 5, the design, the training, and the testing of the MLFFNN-LM and MLFFNN-EBP for predicting the power are presented in detail. Section 6 illustrates and compares the effectiveness and validation of the trained MLFFNN-LM and MLFFNN-EBP using data from the six days which is not used for the training. In Section 7, the obtained results from the proposed method are discussed and compared with other previously published works. Finally, Section 8 presents the work conclusion and some future work.

2. Solar PV Output Power Calculation

To calculate the electrical power obtained from the PV module, the following equations are investigated in [24], [25]:

$$P = \eta_{sc}\tau_g\alpha_{sc}RA[1 - \mu_{sc}(T_{sc} - T_r)] \tag{1}$$

where, η_{sc} is the cell reference efficiency, τ_g is the glass transmissivity, α_{sc} is the solar cell absorptivity, R is the solar radiation (W/m^2), A is the total area of the solar cell (m^2), μ_{sc} is the thermal coefficient of PV cell efficiency $\left(\frac{\%}{°C}\right)$, T_{sc} is the solar cell temperature (°C), and T_r is the reference temperature (°C).

By using MLFFNN, the PV electrical power can be estimated without using the previous equations and depending on the PV system parameters, as discussed below.

3. The Proposed Model Methodology

The MLFFNN is implemented to expect the output power of the solar PV subsystem, which is characterized by strategy simplicity compared to other types of NNs [18], [26], [27]. Furthermore, it is simply and successfully utilized in a variety of problematic fields [28]–[30]. The MLFFNN has been proposed in [31]–[33] owing to its characteristics of adaptivity, parallelism, and generalization. Moreover, it can operate under linear or nonlinear conditions. So, it may require a significant amount of sets of input and aim for the training procedure [34], [35], which can be considered a serious drawback. Here, the proposed work is designed and implemented to treat with this problem and eradicate it. It is observed from Fig. 3, the proposed model strategy can be divided into four stages:

1) Obtaining the original solar PV time series data such as temperature and solar radiation.
2) Data preprocessing in which the data is organized and the missing parameters are initialized.
3) MLFFNN process which is consisted of a training and testing process and the effectiveness process.
4) Visualizing results.

4. MLFFNN Design for Solar PV Output Power Prediction

To precisely design the MLFFNN, the main followed criteria in Ref. [36]–[40] are utilized for exhibiting high performance with the lowest MSE and the TE for the NN inputs. The MLFFNN architecture consists of the input layer, which includes the two inputs, the non-linear (hyperbolic tangent activation function) hidden layer and the output layer.

Firstly, according to the conceived literature review, it is investigated that the distinction between the PV module temperature and the reference temperature $(T_d = T_m - T_r)$, and the radiation (R), have the main influence for performing high MLFFNN execution. Hence, the reference temperature T_r is a constant value equal to 25^oC, and it is abstracted from the module temperature based on the recommendation, as provided in Ref. [25]. In Fig. 4, both inputs of MLFFNN are represented. The hidden layer is responsible for processing the input data and developing the results in the output layer. Here, the output layer estimates the power of the PV power station P'. This estimated power is associated with the realistic one obtained from a real PV power station P.

Secondly, the relationships among the input, hidden, and output layers, as depicted in Fig. 5, are presented with the following equations. The feedforward part of the designed MLFFNN is given as follows:

$$y_j = \varphi_j(h_j) = \varphi_j\left(\sum_{i=0}^{2} w_{ji}\,x_i\right) \tag{2}$$

where x_i are the inputs to the MLFFNN, $x_0 = 1$, $x_1 = T_d(k)$, and $x_2 = R(k)$.

$$\varphi_j(h_j) = \tanh(h_j) \tag{3}$$

$$P' = \varphi_k(O) = \varphi_k\left(\sum_{j=0}^{n} b_{1j}\, y_j\right) = \left(\sum_{j=0}^{n} b_{1j}\, y_j\right) \tag{4}$$

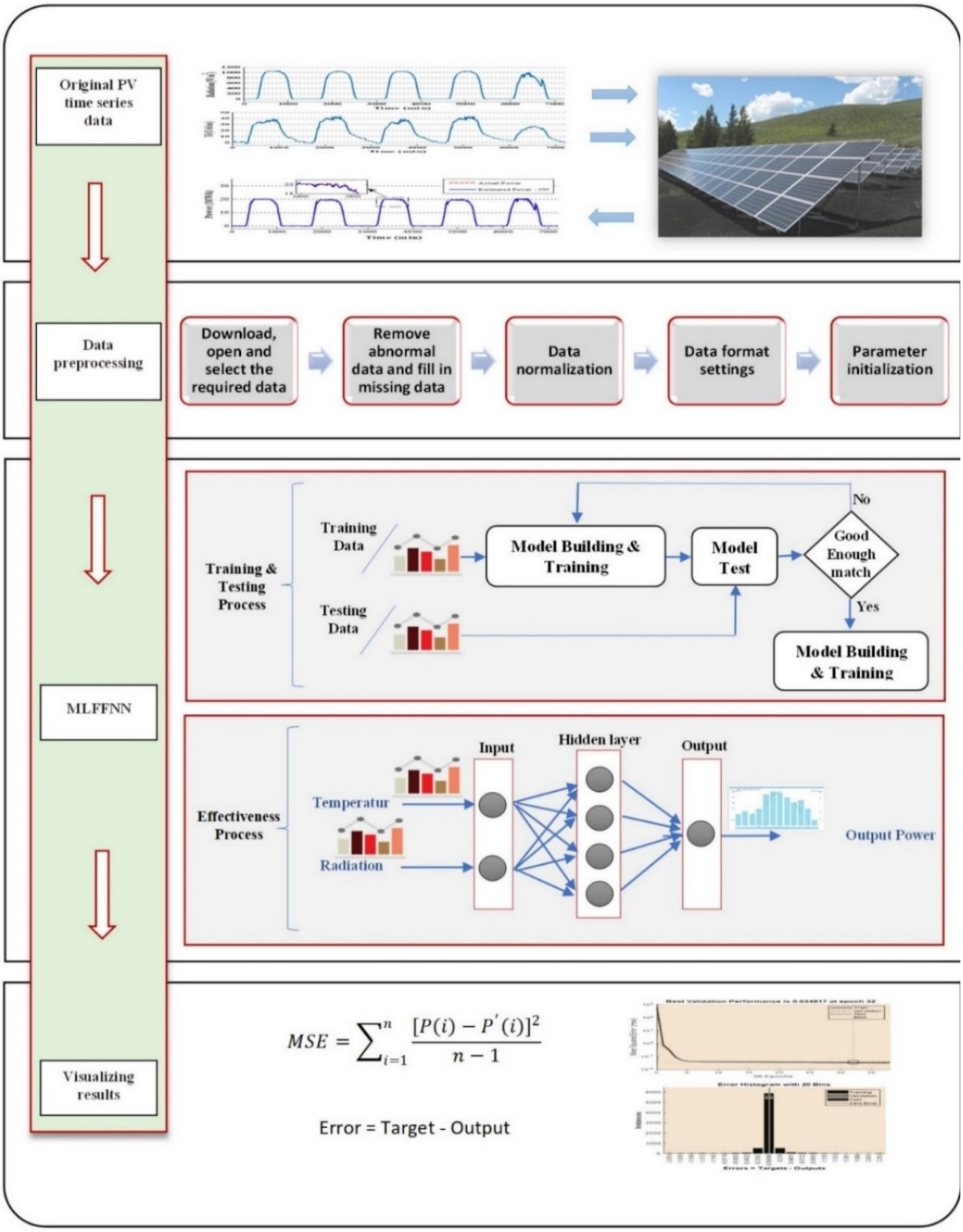
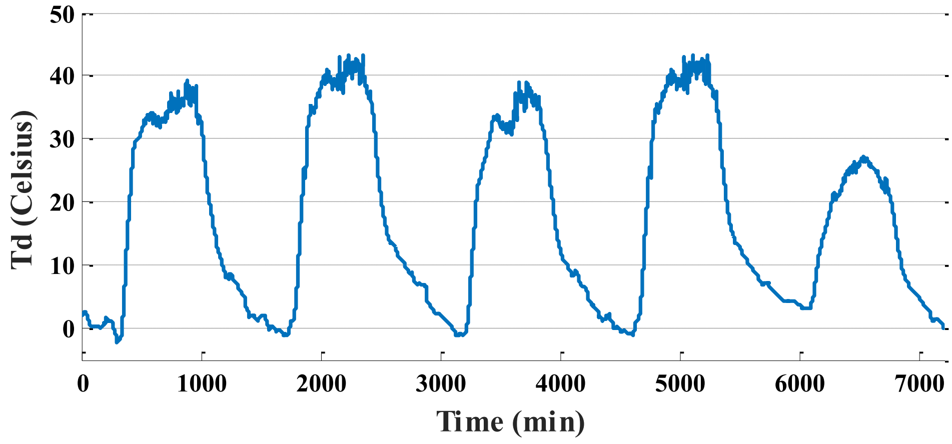
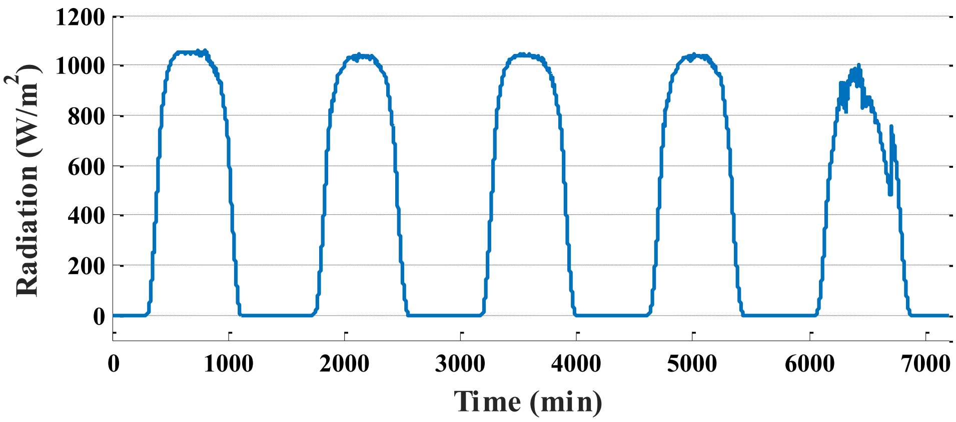


Fig. 3. The proposed model methodology.



(a) The temperature $T_d = T_m - T_r$



(b) The solar radiation (R)

Fig. 4. The inputs of the designed MLFFNN.

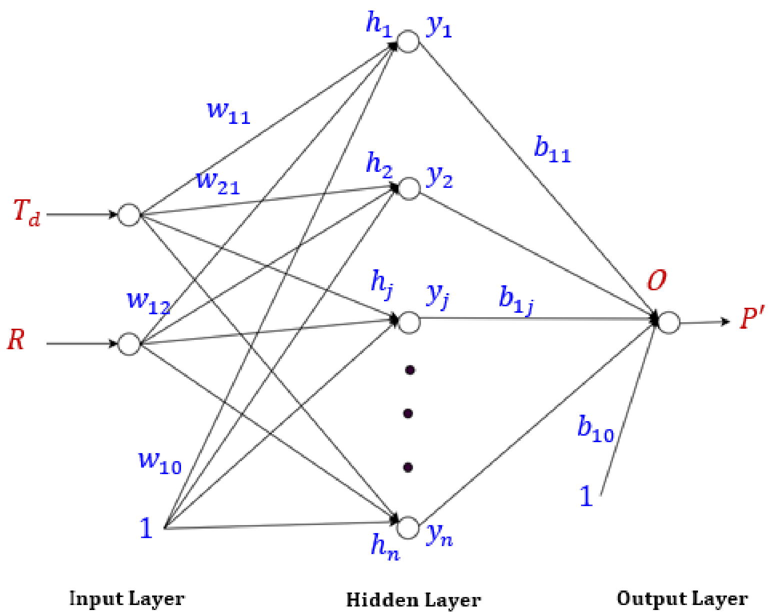


Fig. 5. The design of the proposed MLFFNN structure. The drawing of this architecture is carried out using the online website: https://app.diagrams.net/.

The power P is used only for training the MLFFNN architecture, and the training error $e(t)$ ought to be as tiny as feasible, and it is offered by the subsequent equation:

$$e(t) = P - P' \tag{5}$$

Hence, the training process of the designed MLFFNN is well discussed in the next section.

5. MLFFNN-LM's and MLFFNN-EBP's Training and Testing Processes

In this section, both the training and testing processes of the designed MLFFNN are extensively investigated. During these stages, the following steps are followed with the designed MLFFNN,

1) Import the collected data from the real PV station.
2) Initialize the MLFFNN's Parameters and select the suitable number of hidden neurons.
3) Train the designed MLFFNN.
4) After the training is completed, check the performance of the MLFFNN and the resulting MSE.
5) If the resulting MSE is high value and not satisfactory, go again to step 2.
6) If the resulting MSE is very small and close to zero (satisfactory),
   - 5.1 Test the trained MLFFNN by using the same data that was used for training and check the training/approximation error.
   - 5.2 If this training/approximation error is low and satisfactory, go to step 7.
   - 5.3 If this training/approximation error is high and not satisfactory, go again to step 2.
7) Check the generalization ability/effectiveness of the trained MLFFNN by using different data than the ones used for training.
8) The trained MLFFNN is ready for PV output power prediction.

These steps are presented in a flowchart, as shown in Fig. 6. During the next subsections and section 6, all these steps are discussed and investigated in detail.

5.1. Training Procedure

For training the designed MLFFNN, two learning algorithms are utilized; LM and EBP. The properties of these algorithms are discussed as follows.

LM algorithm has the following properties. This algorithm can easily process the data in a fast way, which is considered a second-order optimization algorithm that has the ability for vast convergence based on Newton's Method [41], [42]. Compared to other learning algorithms, LM learning has the exchange-off between the rapid-learning speed of the conventional Newton's process and the definite convergence of the ascent slope [41], [43]. This learning is proper for enormous datasets along with converges in fewer iterations and in a tiny time. The familiar weights of the MLFFNN via the LM algorithm are given by the next equation [18], [27]:

$$w_{k+1} = w_k - [\mathrm{H} + \lambda\mathrm{I}]^{-1} g \tag{6}$$

where H and g are Hessian and the gradient vector of the second-order function, respectively, I is the identity matrix of the same dimensions as H, and λ is a regularizing or loading parameter that forces the sum matrix $(\mathrm{H} + \lambda\mathrm{I})$ to be positive definite, and safely well-conditioned throughout the computation.

EBP has the following properties. This algorithm is widely used because it is simple to implement [27], [44]. In addition, it has accelerating convergence to diminish the error function if the appropriate values of the learning value and the momentum constant are employed [18], [45]. In general, the training using EBP takes longer time than the LM algorithm [27]. The adjusted weights of the MLFFNN using the EBP algorithm are given by the next equation [27], [46]:

$$w_{k+1} = \alpha w_K + \Delta w = \alpha w_K + \eta \delta x \quad (7)$$

where α is the momentum constant which is a positive number ($0 \leq \alpha < 1$), H is the learning rate parameter, δ is the local gradient, and x is the input signal of the neuron.

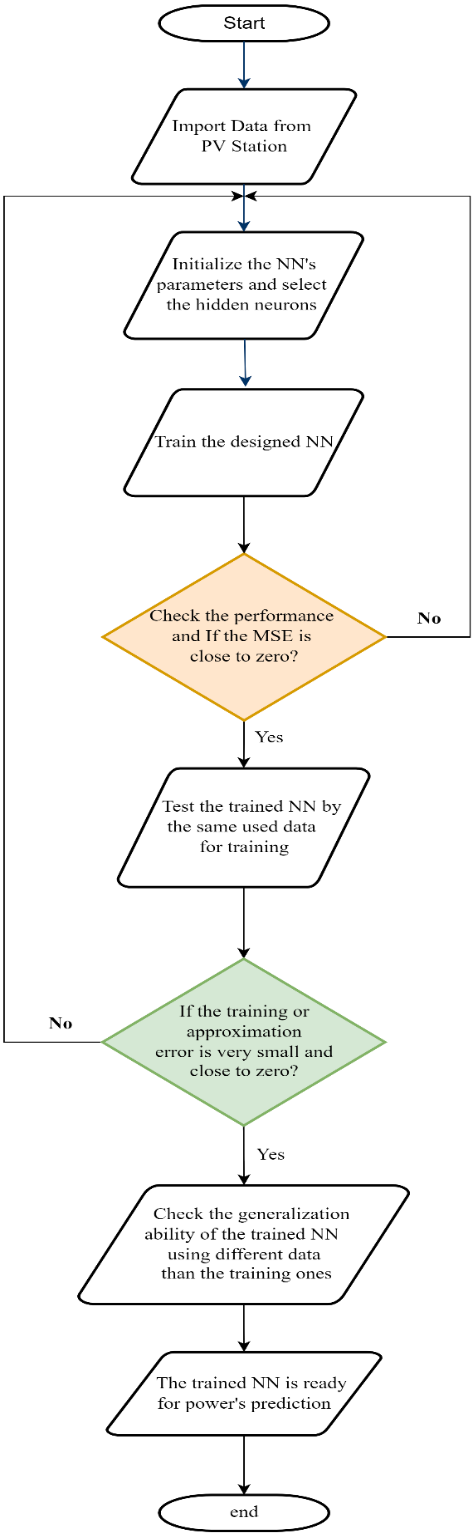


Fig. 6. A flowchart illustrates the following steps during the stages of training, testing, and effective investigation of the designed MLFFNN. The flowchart is drawn using the online website: https://app.diagrams.net/.

The data used for training the MLFFNN-LM and MLFFNN-EBP are obtained from a real PV power station in Egypt. The collected data are for six days. The data for five days are used for training (see Fig. 4.), whereas the data for the sixth day are used for checking the effectiveness of the trained MLFFNN. The full number of input-output pairs of the data utilized for training is 7200. From these statistics, 90% are employed for the training process, 5% for the authorization method, and 5% for assessing performance. After attempting numerous various weights' initializations and a number of hidden neurons, the best parameters of the MLFFNN-LM and MLFFNN-EBP that realize the superior performance are offered in Table 1.

Table 1. The best parameters of the MLFFNN-LM and MLFFNN-EBP achieve high performance.

Parameter	MLFFNN-EBP	MLFFNN-LM
Number of hidden neurons	70	70
Epochs/Iterations	1000	32
Training time	2.20 min.	0.9 min.
Lowest MSE	0.0238	0.034817

The results obtained from the training process of both MLFFNN-LM and MLFFNN-EBP, such as the training MSE and the regression, are presented in Fig. 7 and Fig. 8, respectively. As clear from Table 1 and Fig. 7, and Fig. 8, the obtained MSE is very low and close to the value of zero. The regression measures the correlation between the estimated power by the MLFFNN and the actual power. The regression (Reg) is close to 1, which means that the convergence/approximation between the two powers (P, P') is very good. These results prove that the MLFFNN-LM and MLFFNN-EBP are trained very well, and they are ready to predict the PV power correctly. The obtained MSE by MLFFNN-EBP is slightly better/lower than the MSE by MLFFNN-LM. The obtained regression is approximately the same. However, the training time and the iterations are higher in the case of using MLFFNN-EBP. Indeed, the training is occurring offline, and therefore the training time is not very valuable and important because the major aim is to obtain a very well-trained NN that can predict the output power efficiency more correctly.

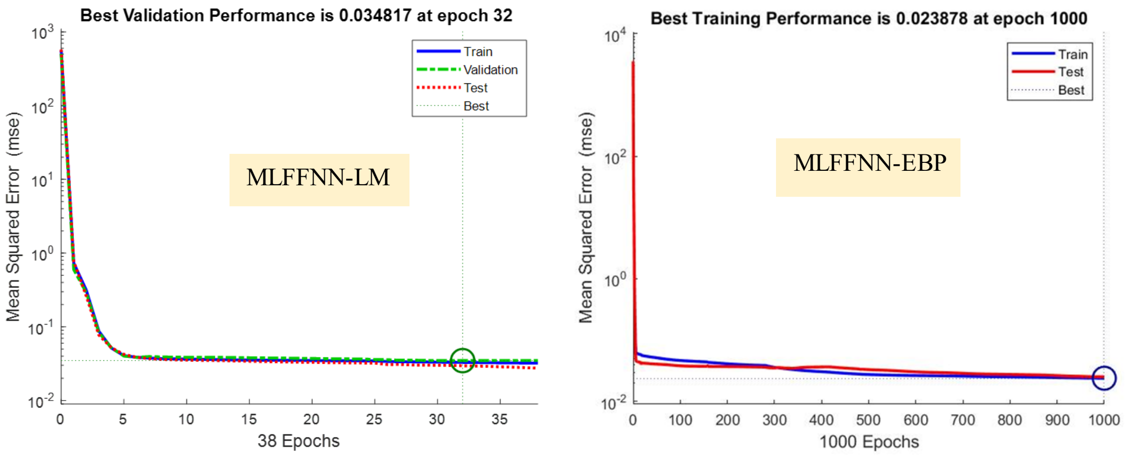


Fig. 7. The obtained MSE during the training of the MLFFNN-LM and MLFFNN-EBP.

5.2. Testing Procedure

Once the training of the MLFFNN-LM and MLFFNN-EBP is finished completely, these trained NNs are checked and investigated with the identical dataset that was applied for the training process to make an insight into the approximation. The approximation error between the estimated power P' by the NN (whether MLFFNN-LM or MLFFNN-EBP) and the actual one obtained from real PV power station P is presented in Fig. 9. In addition, the average, maximum, minimum, and standard deviation (std.) of the absolute value of this approximation error using both cases are presented in Table 2.

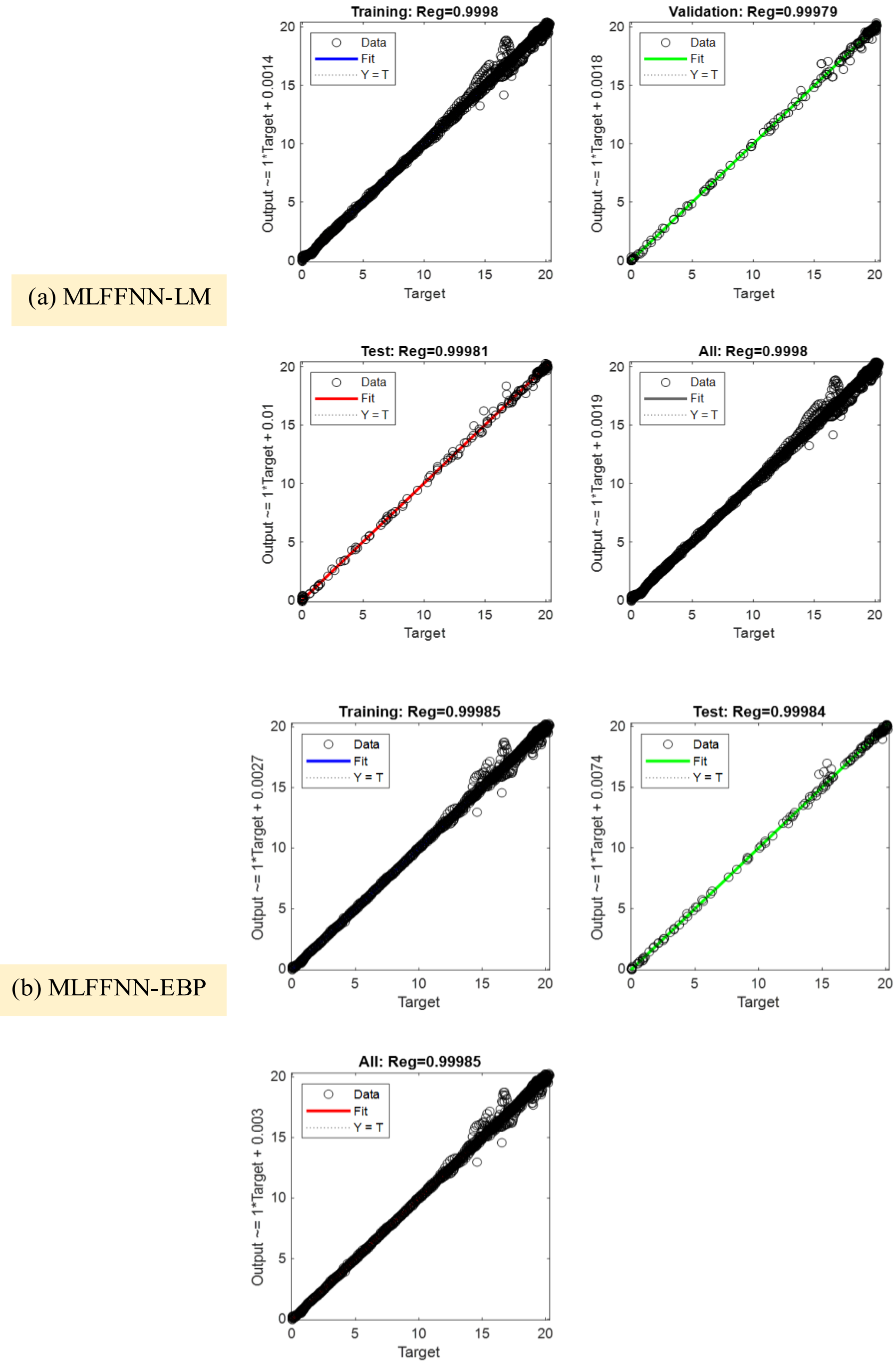


Fig. 8. The obtained regression from the training of the MLFFNN-LM and MLFFNN-EBP.

Table 2. The average, maximum, and std. of the approximation error using MLFFNN-LM and MLFFNN-EBP.

Parameter	MLFFNN-EBP	MLFFNN-LM
Average of absolute error (MWh)	0.0607	0.0779
Std. of absolute error	0.1424	0.1636
Maximum of absolute error (MWh)	2.0436	2.3754
Minimum of absolute error (MWh)	3.1785e-06	7.3231e-06

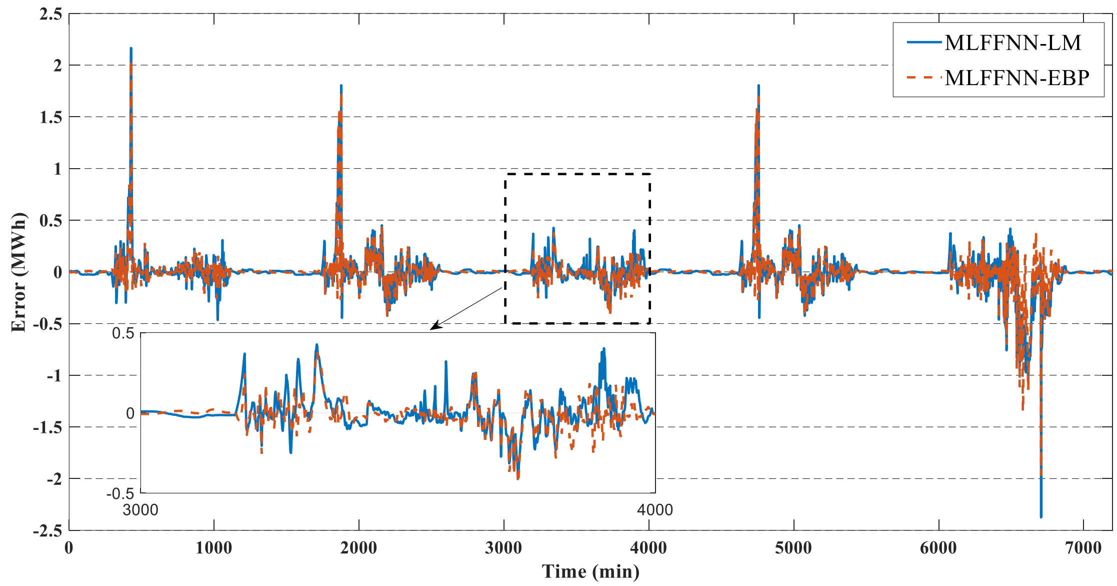


Fig. 9. The approximation error between the estimated power P' and the actual one P using the MLFFNN.

It is clear from Fig. 9 and Table 2 that the approximation error between the estimated power by the NN and the actual power is low, which means that the NN is trained very well. The approximation error in the instance of using MLFFNN-EBP is slightly lower and better than the one in the case of using MLFFNN-LM. The approximation or the convergence between the estimated power by the MLFFNN-LM and the actual power is shown in Fig. 10. There is no need to present the case of using MLFFNN-EBP as it gives approximately the same results and shape.

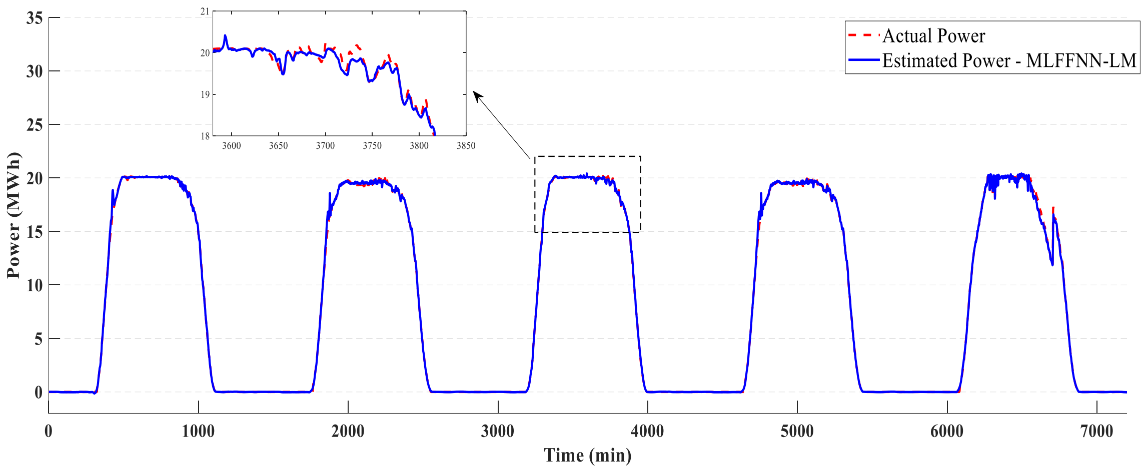


Fig. 10. The comparison between the estimated power by MLFFNN-LM and the actual one obtained from the real PV power station. The same results are obtained by MLFFNN-EBP.

6. MLFFNN-LM's and MLFFNN-EBP's Validation and Assessment for Day-Ahead Forecasting

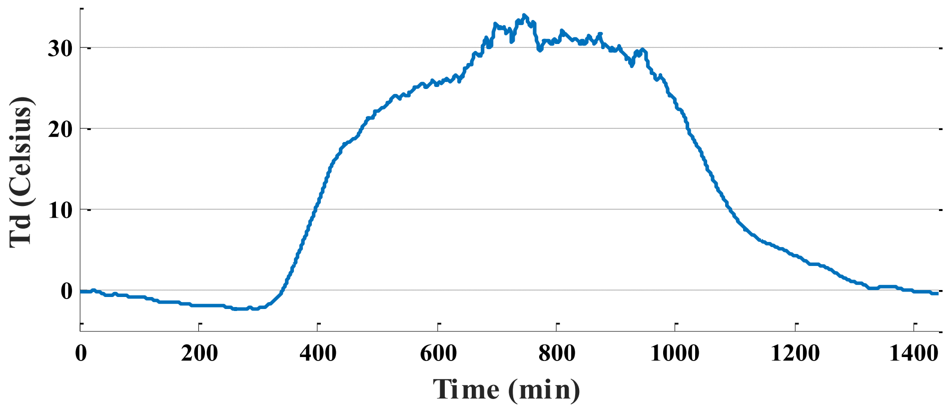
In this section, the trained MLFFNN-LM and MLFFNN-EBP are assessed via dissimilar data than the data applied for the training process. Hence, the data of the sixth day obtained from the real PV power station are used to check the effectiveness and the generalization capability of the trained MLFFNN. These data (temperature T_d and solar radiation R which are the inputs to the NN) are presented in Fig. 11.

For the assessment of the MLFFNN-LM's and MLFFNN-EBP's performance, the comparisons between the estimated power by them and the actual one on the sixth day are presented in Fig. 12 and Fig. 13. In addition, Table 3 illustrates the average, maximum, minimum, and std of the absolute error between the two powers in both cases.

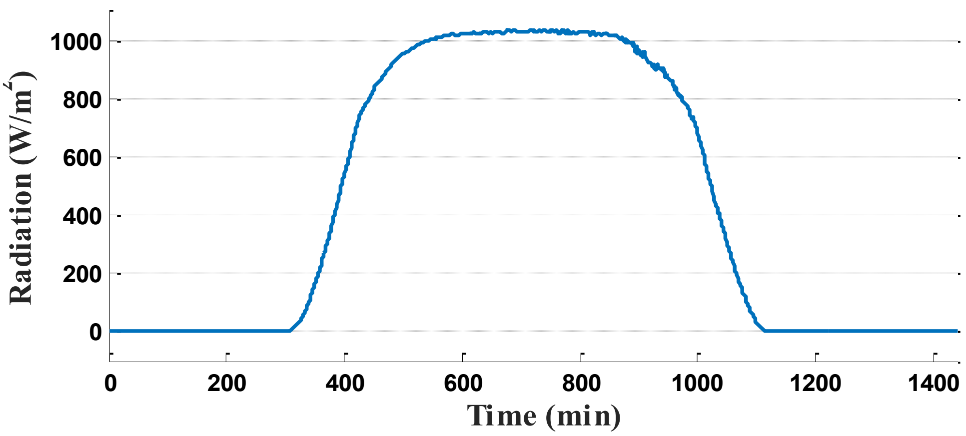
Table 3. The average, maximum, and std. of the absolute error obtained by MLFFNN-LM and by MLFFNN-EBP and using different data than the data used for the training process.

Parameter	MLFFNN-EBP	MLFFNN-LM
Average of absolute error (MWh)	0.2333	0.2842
Std. of absolute error	0.5205	0.5423
Maximum of absolute error (MWh)	3.8123	3.2880
Minimum of absolute error (MWh)	5.2968e-05	4.3581e-05

As shown in Fig. 12, Fig. 13, and Table 3, the error between the estimated power by the trained MLFFNN and the actual power is, to some extent, low and satisfactory but is higher compared with the training case presented in Subsection 5.2. Indeed, this is logical because, in the current case, the used data is different than the training data. This proves that the MLFFNN is trained very well and can work and generalize well under various circumstances and data than the ones utilized for training. The error obtained by the MLFFNN-LM is slightly better and lower compared with the case which uses MLFFNN-EBP, and this is clear from Fig. 13. This shows that the effectiveness and the generalization ability using MLFFNN-LM are slightly better compared with using MLFFNN-EBP.



(a) The temperature $T_d = T_m - T_r$



(b) The solar radiation (R).

Fig. 11. The data of the sixth day was used to check the effectiveness of the trained MLFFNN. These data are the input of the NN.

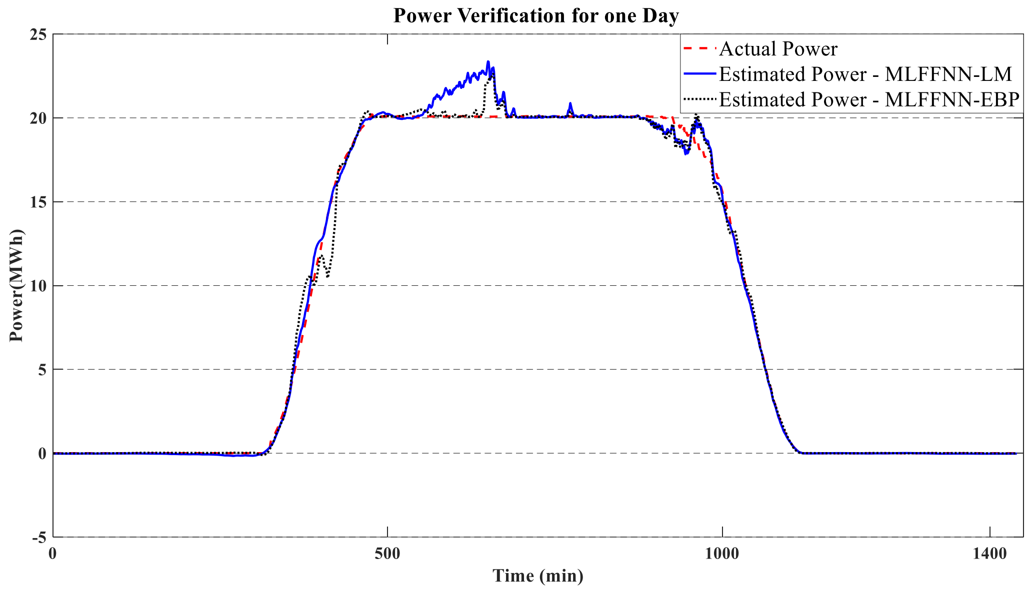


Fig. 12. The comparison between the estimated power by trained MLFFNN-LM and MLFFNN-EBP and the actual one, using data different than the data utilized for the training process.

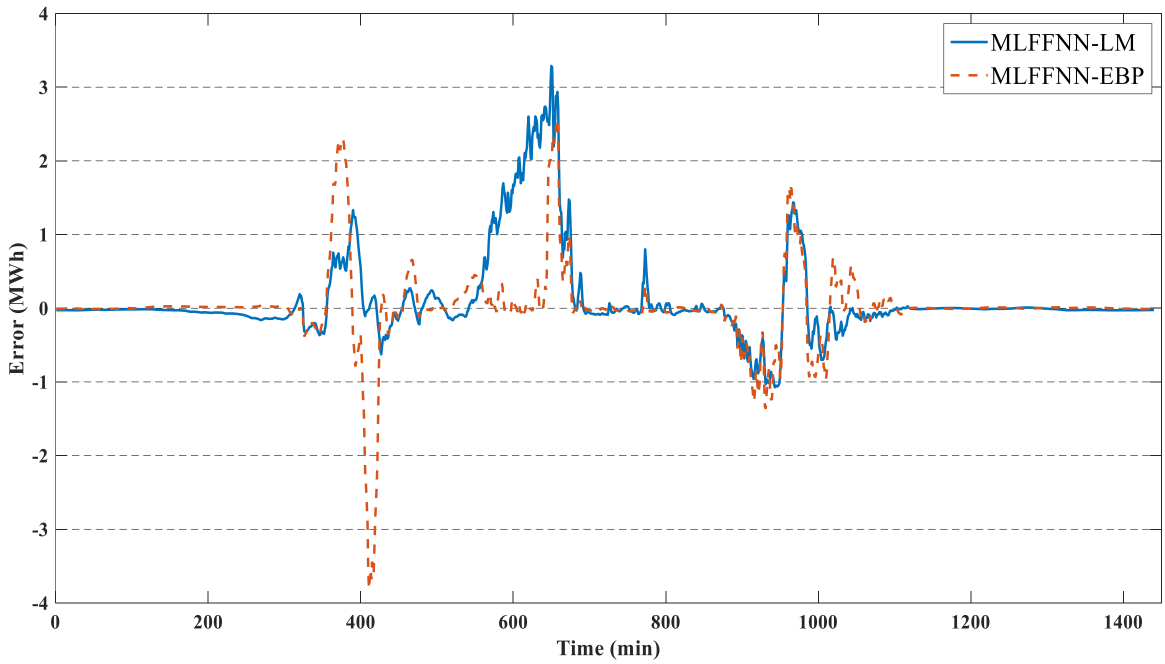


Fig. 13. The error between the estimated power by trained MLFFNN and the actual one, using data different than the data applied for the training process. The error is presented in the case of using MLFFNN-LM and MLFFNN-EBP.

7. Discussion and Comparison

In this section, the obtained results by the proposed MLFFNN-LM and MLFFNN-EBP are discussed. In addition, they are compared with other previously published research works.

The presented results in Section 5 and Section 6 show that the performance of the MLFFNN-EBP is slightly better than the performance of the MLFFNN-LM in the training stage because the MLFFNN-EBP achieves the lower MSE and the lower training/approximation error. However, it takes a longer time for training since it needs more iterations/epochs as well as time. The training time is not very significant because it occurs offline as well as the main objective is to obtain a very well-trained NN that can work and generalize in various situations and cases. The generalization ability and the effectiveness of the trained NN are investigated by handling several different data from the training data, and the MLFFNN-LM achieves a slightly better performance (lower error) associated with the MLFFNN-EBP. This indicates that the MLFFNN-EBP affects by different conditions and cases. As this stage is the very important one, therefore we recommend using the MLFFNN-LM for forecasting the output power of the PV power station. However, all of them can be used as the difference is small between them.

Our proposed MLFFNN-LM and MLFFNN-EBP are compared with other previously published methods, which are used for short-term power prediction and presented in Ref. [15], [17], [20], [22], [23]. This comparison is presented in Table 4 in terms of the following parameters (1) The used algorithm; (2) The size of the input layer and the inputs' parameters; (3) The number of hidden neurons if found; (4) The MSE, RMSE, and MAE, and; (5) Generalization ability investigation (if verified and checked or not).

Table 4. Comparing the proposed method in this paper with other previously published methods in terms of different factors and parameters.

Reference	Algorithm	Inputs and Their number	Number of Hidden Neurons	MSE	RMSE	MAE	Generalization Ability
Wang et al. (2016) [15]	The partial functional linear regression model	Four inputs: (Pressure, relative humidity, temperature, wind speed)	----------	0.348	0.59	0.1134	-----------
Kumar et al. (2021) [20]	Elman NN	Four inputs: (Relative humidity, ambient temperature, sky image, solar irradiance)	10	0.1101	0.29	0.18	-----------
	FFNN		10	0.1124	0.31	0.19	
	Generalized regression NN		2440	0.087	0.32	0.23123	
Alomari et al. (2018) [22]	MLFFNN	Five inputs: (Solar irradiance of the previous five days)	22	0.0053	0.0721	---------	-----------
Alshafeey et al. (2021) [23]	Multiple regression	Six inputs: (Air temperature, cloud capacity, irradiation, humidity, precipitable water, snow depth)	4, 32, 35	2234	47	26	----------
	MLFFNN			2171	46	24	
Yen et al. (2021) [17]	Support Vector Machine (SVM)	Four inputs: (Temperature, humidity, rainfall, and wind speed)	----------	--------	2.5094	1.6875	----------
	Random Forest		----------	--------	1.5878	1.0096	
The proposed method in this paper	MLFFNN-EBP	Two inputs: (Module temperature and solar radiation)	70	0.0238	0.1543	---------	Checked and verified
	MLFFNN-LM		70	0.0348	0.1870	---------	

It is clear from Table 4 that our proposed method (whether MLFFNN-LM or MLFFNN-EBP) and the method presented in [22] achieve the lowest MSE and RMSE compared with the other methods. This means that our method and the one presented in [22] have the highest accuracy of the PV power prediction compared with the other methods. Furthermore, our method has only two inputs compared with the other methods, which have four or more four inputs. This proves that the size of the input layer of our method is the smallest compared with others. Therefore, the complexity of our proposed method is lower. The generalization ability and the effectiveness under different conditions and cases are checked and verified only by our method. Finally, we conclude that our proposed method is efficient in predicting the solar PV output power correctly.

8. Conclusion and Future Work

In this article, the MLFFNN is intended to predict the output power of a solar PV power substation. Hence, the module temperature and the solar radiation are utilized as its input in order to estimate the solar PV power as an output. The accurate power prediction using the trained MLFFNN can help to avoid the fall of the power that maybe happen at any time. Hence, data of six days are collected from a real solar PV power station in Egypt. From this, the data of the first five days are applied for the training, which occurred using LM and EBP algorithms. The results from the training and testing processes' performance show that the MLFFNN-EBP has slightly better performance (lowest MSE and training error) than the ones obtained by MLFFNN-LM. The data of the sixth day, which are not used for the training, are used to check and investigate the trained MLFFNN-LM and MLFFNN-EBP. From this process, a slightly better performance (lower error) is obtained by the trained MLFFNN-LM, which means that the MLFFNN-EBP is affected by the different conditions. Both MLFFNN-LM and MLFFNN-EBP are working very well and efficiently to predict the power correctly. However, we recommend using the MLFFNN-LM, which has better effectiveness. Our proposed method is compared with other previously published methods by researchers. From this comparison, it is concluded that our method has the highest accuracy in PV output power prediction. In addition, it is the simplest method, and the only one in its generalization ability and effectiveness are checked and verified.

The good obtained results in this work motivate us to make further investigation of MLFFNN-LM and MLFFNN-EBP for medium- and long- terms prediction of the PV power station. Investigating different types of NN, such as the recurrent NN, radial basis function NN, cascaded forward NN, and so on, is recommended for predicting the output power. Deep learning can also be used. In addition, more data than the used ones in this work can be collected and used.

Author Contribution: Conceptualization, Abdel-Nasser Sharkawy, and Mustafa M. Ali; methodology, Abdel-Nasser Sharkawy, and Mustafa M. Ali; software, Abdel-Nasser Sharkawy; validation, Abdel-Nasser Sharkawy; formal analysis, Abdel-Nasser Sharkawy, Mustafa M. Ali, Hossam H. H. Mousa; investigation, Abdel-Nasser Sharkawy, and Mustafa M. Ali; resources, Abdel-Nasser Sharkawy and Mustafa M. Ali; data curation, Mustafa M. Ali; writing—original draft preparation, Abdel-Nasser Sharkawy, Mustafa M. Ali, Hossam H. H. Mousa; writing—review and editing, Abdel-Nasser Sharkawy, Mustafa M. Ali, Hossam H. H. Mousa, Ahmed S. Ali, G. T. Abdel-Jaber; visualization, Abdel-Nasser Sharkawy; supervision, Ahmed S. Ali and G. T. Abdel-Jaber. All authors read and approved the final version of the paper.

Funding: This research received no external funding.

Acknowledgment: The authors thank their colleagues who provided insight and expertise that greatly assisted the research.

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

Supplementary Materials: Not Applicable.

References

[1] J. L. Gómez, A. O. Martínez, F. T. Pastoriza, L. F. Garrido, E. G. Álvarez, and J. A. O. García, "Photovoltaic power prediction using artificial neural networks and numerical weather data," *Sustain.*, vol. 12, no. 24, pp. 1–19, 2020, https://doi.org/10.3390/su122410295.

[2] Z. A. Alrowaili *et al.*, "Robust adaptive HCS MPPT algorithm-based wind generation system using model reference adaptive control," *Sensors*, vol. 21, no. 15, pp. 1–20, 2021, https://doi.org/10.3390/s21155187.

[3] M. M. Ali, A. R. Youssef, A. S. Ali, and G. T. Abdel-Jaber, "Variable step size PO MPPT algorithm using model reference adaptive control for optimal power extraction," *Int. Trans. Electr. Energy Syst.*, vol. 30, no. 1, pp. 1–21, 2020, https://doi.org/10.1002/2050-7038.12151.

[4] "In 2019 the solar PV market increased an estimated 12% to around 115 GW," 2020. [Online]. Available: https://www.evwind.es/2020/07/05/in-2019-the-solar-pv-market-increased-an-estimated-12-to-around-115-gw/75561.

[5] S. K. Prajapati and K. B. Sahay, "A Survey Paper on Solar Irradiance Forecasting Methods," *Int. J. Eng. Res.*, vol. 5, no. 03, pp. 536–541, 2016, https://www.ijert.org/research/a-survey-paper-on-solar-irradiance-forecasting-methods-IJERTV5IS030469.pdf.

[6] M. N. Akhter, S. Mekhilef, H. Mokhlis, and N. M. Shah, "Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques," *IET Renew. Power Gener.*, vol. 13, no. 7, pp. 1009–1023, 2019, https://doi.org/10.1049/iet-rpg.2018.5649.

[7] U. K. Das *et al.*, "Forecasting of photovoltaic power generation and model optimization: A review," *Renew. Sustain. Energy Rev.*, vol. 81, no. April 2017, pp. 912–928, 2018, https://doi.org/10.1016/j.rser.2017.08.017.

[8] C. Voyant *et al.*, "Machine learning methods for solar radiation forecasting: A review," *Renew. Energy*, vol. 105, pp. 569–582, 2017, https://doi.org/10.1016/j.renene.2016.12.095.

[9] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, 2015, https://doi.org/10.1016/j.rser.2015.04.065.

[10] S. Sobri, S. Koohi-Kamali, and N. A. Rahim, "Solar photovoltaic generation forecasting methods: A review," *Energy Convers. Manag.*, vol. 156, pp. 459–497, 2018, https://doi.org/10.1016/j.enconman.2017.11.019.

[11] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de-Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," *Sol. Energy*, vol. 136, pp. 78–111, 2016, https://doi.org/10.1016/j.solener.2016.06.069.

[12] A. Dolara, F. Grimaccia, S. Leva, M. Mussetta, and E. Ogliari, "A physical hybrid artificial neural network for short term forecasting of PV plant power output," *Energies*, vol. 8, no. 2, pp. 1138–1153, 2015. https://doi.org/10.3390/en8021138.

[13] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A Review of Wind Power and Wind Speed Forecasting Methods With Different Time Horizons," in *North American Power Symposium 2010*, 2010, pp. 1–8. https://doi.org/10.1109/NAPS.2010.5619586.

[14] M. Kumar, N. Kumar, and S. S. Chandel, "power prediction of photovoltaic system using four parameter model," *Int. J. Electr. Electron. Data Commun.*, vol. 3, no. 5, pp. 82–86, 2015, http://www.iraj.in/journal/journal_file/journal_pdf/1-133-143140928582-86.pdf.

[15] G. Wang, Y. Su, and L. Shu, "One-day-ahead daily power forecasting of photovoltaic systems based on partial functional linear regression models," *Renew. Energy*, vol. 96, pp. 469–478, 2016, https://doi.org/10.1016/j.renene.2016.04.089.

[16] K. R. Müller, A. J. Smoła, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1997, vol. 1327, pp. 999–1004, https://alex.smola.org/papers/1997/MulSmoRatSchetal97.pdf.

[17] C. F. Yen, H. Y. Hsieh, K. W. Su, M. C. Yu, and J. S. Leu, "Solar Power Prediction via Support Vector Machine and Random Forest," *E3S Web Conf.*, vol. 69, no. 2, pp. 1–6, 2018, https://doi.org/10.1051/e3sconf/20186901004.

[18] S. Haykin, *Neural Networks and Learning Machines*, Third Edit. Pearson, 2009, https://cours.etsmtl.ca/sys843/REFS/Books/ebook_Haykin09.pdf.

[19] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015. http://neuralnetworksanddeeplearning.com/.

[20] P. M. Kumar, R. Saravanakumar, A. Karthick, and V. Mohanavel, "Artificial neural network-based output power prediction of grid-connected semitransparent photovoltaic system," *Environ. Sci. Pollut. Res.*, 2021, https://doi.org/10.1007/s11356-021-16398-6.

[21] S. H. Oudjana, A. Hellal, and I. H. Mahamed, "Short term photovoltaic power generation forecasting using neural network," in *2012 11th International Conference on Environment and Electrical Engineering*, 2012, pp. 706–7011, https://doi.org/10.1109/EEEIC.2012.6221469.
[22] M. H. Alomari, J. Adeeb, and O. Younis, "Solar photovoltaic power forecasting in Jordan using artificial neural networks," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 1, pp. 497–504, 2018, http://doi.org/10.11591/ijece.v8i1.pp497-504.
[23] M. AlShafeey and C. Csáki, "Evaluating neural network and linear regression photovoltaic power forecasting models based on different input methods," *Energy Reports*, vol. 7, pp. 7601–7614, 2021, https://doi.org/10.1016/j.egyr.2021.10.125.
[24] A. Nahar, M. Hasanuzzaman, and N. A. Rahim, "Numerical and experimental investigation on the performance of a photovoltaic thermal collector with parallel plate flow channel under different operating conditions in Malaysia," *Sol. Energy*, vol. 144, pp. 517–528, 2017, https://doi.org/10.1016/j.solener.2017.01.041.
[25] R. Nasrin, M. Hasanuzzaman, and N. A. Rahim, "Effect of high irradiation on photovoltaic power and energy," *Int. J. Energy Res.*, vol. 42, no. 3, pp. 1115–1131, 2018, https://doi.org/10.1002/er.3907.
[26] J. Schmidhuber, "Deep learning in neural networks : An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, https://doi.org/10.1016/j.neunet.2014.09.003.
[27] A.-N. Sharkawy, "Principle of Neural Network and Its Main Types : Review," *J. Adv. Appl. Comput. Math.*, vol. 7, pp. 8–19, 2020, https://doi.org/10.15377/2409-5761.2020.07.2.
[28] S. C. Chen, S. W. Lin, T. Y. Tseng, and H. C. Lin, "Optimization of back-propagation network using simulated annealing approach," in *2006 IEEE International Conference on Systems, Man and Cybernetics*, 2006, pp. 2819–2824, https://doi.org/10.1109/ICSMC.2006.385301.
[29] M. A. Sassi, M. J. D. Otis, and A. Campeau-Lecours, "Active stability observer using artificial neural network for intuitive physical human–robot interaction," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 4, pp. 1–16, 2017, https://doi.org/10.1177%2F1729881417727326.
[30] E. De Momi, L. Kranendonk, M. Valenti, N. Enayati, and G. Ferrigno, "A Neural Network-Based Approach for Trajectory Planning in Robot–Human Handover Tasks," *Front. Robot. AI*, vol. 3, no. June, pp. 1–10, 2016, https://doi.org/10.3389/frobt.2016.00034.
[31] A. B. Rad, T. W. Bui, V. Li, and Y. K. Wong, "A new on-line pid tuning method using neural networks," *IFAC Proc. Vol. IFAC Work. Digit. Control Past, Present Futur. PID Control*, vol. 33, no. 4, pp. 443–448, 2000, https://doi.org/10.1016/S1474-6670(17)38283-6.
[32] S. A. Elbelady, H. E. Fawaz, and A. M. A. Aziz, "Online Self Tuning PID Control Using Neural Network for Tracking Control of a Pneumatic Cylinder Using Pulse Width Modulation Piloted Digital Valves," *Int. J. Mech. Mechatronics Eng. IJMME-IJENS*, vol. 16, no. 3, pp. 123–136, 2016, http://ijens.org/Vol_16_I_03/163603-9898-IJMME-IJENS.pdf.
[33] R. Hernández-Alvarado, L. G. García-Valdovinos, T. Salgado-Jiménez, A. Gómez-Espinosa, and F. Fonseca-Navarro, "Neural Network-Based Self-Tuning PID Control for Underwater Vehicles," *sensors*, vol. 16, no. 9: 1429, pp. 1–18, 2016, https://doi.org/10.3390/s16091429.
[34] P. Jeatrakul and K. W. Wong, "Comparing the performance of different neural networks for binary classification problems," in *2009 8th International Symposium on Natural Language Processing, SNLP '09*, 2009, pp. 111–115, https://doi.org/10.1109/SNLP.2009.5340935.
[35] D. Anderson and G. McNeill, "Artificial neural networks technology: A DACS state-of-the-art report," Utica, New York, 1992, https://dokumen.tips/documents/artificial-neural-networks-technology-neural-networks-technology-a-dacs-state-of-the-art.html?page=2.
[36] A.-N. Sharkawy and N. Aspragathos, "Human-Robot Collision Detection Based on Neural Networks," *Int. J. Mech. Eng. Robot. Res.*, vol. 7, no. 2, pp. 150–157, 2018, https://doi.org/10.18178/IJMERR.7.2.150-157.
[37] A.-N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Manipulator Collision Detection and Collided Link Identification based on Neural Networks," in *Advances in Service and Industrial Robotics. RAAD 2018. Mechanisms and Machine Science*, A. Nikos, K. Panagiotis, and M. Vassilis, Eds. Springer, Cham, 2018, pp. 3–12, https://doi.org/10.1007/978-3-030-00232-9_1.
[38] A. N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Neural Network Design for Manipulator Collision Detection Based only on the Joint Position Sensors," *Robotica*, vol. 38, no. Special Issue 10: Human–Robot Interaction (HRI), pp. 1737–1755, 2020, https://doi.org/10.1017/S0263574719000985.
[39] A. N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network," *Soft Comput.*, vol. 24, no. 9, pp. 6687–6719, 2020, https://doi.org/10.1007/s00500-019-04306-7.
[40] A. N. Sharkawy and A. A. Mostfa, "Neural networks' design and training for safe human-robot

cooperation," *J. King Saud Univ. - Eng. Sci.*, pp. 1–15, 2021. https://doi.org/10.1016/j.jksues.2021.02.004
[41] K. Du and M. N. S. Swamy, *Neural Networks and Statistical Learning*. Springer, 2014, https://doi.org/10.1007/978-1-4471-7452-3.
[42] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *J. Soc. Ind. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963, https://doi.org/10.1137/0111030.
[43] M. T. Hagan and M. B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Trans. NEURAL NETWORKS*, vol. 5, no. 6, pp. 2–6, 1994, https://doi.org/10.1109/72.329697.
[44] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Exploration of the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362, https://ieeexplore.ieee.org/document/6302929.
[45] R. Rojas, *Neural Networks - A Systematic Introduction*. Berlin: Springer-Verlag, 1996, https://page.mi.fu-berlin.de/rojas/neural/.
[46] A. N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "A neural network-based approach for variable admittance control in human–robot cooperation: online adjustment of the virtual inertia," *Intell. Serv. Robot.*, vol. 13, no. 4, pp. 495–519, 2020, https://doi.org/10.1007/s11370-020-00337-4.