International Journal of Engineering & Technology, 8 (1.7) (2019) 126-132

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Improved Artificial Neural Network for Grid-Connected Photovoltaic System Output Prediction

Shahril Irwan Sulaiman[1,2]*, Norfarizani Nordin[1] and Ahmad Maliki Omar[1,2]

[1] *Green Energy Research Centre (GERC), Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia*
[2] *Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia*
**Corresponding author E-mail: shahril_irwan2004@yahoo.com*

Abstract

This paper presents the output prediction of Grid-Connected Photovoltaic (GCPV) system using a multi-layer feedforward neural network. Conventional prediction requires mathematical expressions that need to be updated whenever new system is investigated. However, the introduction of Artificial Neural Network (ANN) in this study eliminated the need for using mathematical expressions. The MLFNN inputs were set to be Solar Irradiance (SI), Ambient Temperature (AT) and Module Temperature (MT) with the respective both current and previous five-minute values while the sole output was set to be the AC power from the GCPV system. The MLFNN was implemented in two stages, i.e. the training and testing. The results showed that the MLFNN model had outperformed the existing MLFNN model using SI and AT as inputs without previous five-minute values in producing the lowest Root Mean Square Error (RMSE) and highest correlation coefficient during both training and testing processes.

Keywords: *photovoltaic; multi-layer feedforward neural network; solar irradiance; ambient temperature; module temperature*.

1. Introduction

The electricity demand has seen a tremendous increase due to the rapid economic growth worldwide. This demand is mainly associated to the needs of various industries besides the household consumption. The global electricity mainly produced from fossil fuel based resources followed by renewable energy resources and nuclear energy [1]. Although the fossil fuel-based electricity generation has been consistently dominating the world electricity market, the issue of global warming due to the uncontrolled release of greenhouse gas (GHG) emissions remains the primary concern when using such technology. On the other hand, nuclear power offers an alternative mode for large-scale electricity generation. However, this technology is often seen as unsafe resources and very dangerous [2]. The threat of nuclear disasters and the problem with the disposal of radioactive waste is another issue that gets a lot of attention [3].
Another option for electricity generation is by using renewable energy (RE) resources such as biomass, wind, solar, hydropower and geothermal [4, 5]. Unlike fossil fuels and nuclear, RE are deemed to be more environmental friendly with the fuel resources being freely available. Examples of RE technology are wind power, geothermal power, hydropower, biomass and photovoltaic. Nevertheless, according to Global Status Report REN 21 [6], solar photovoltaic (PV) has been one of the fastest growing RE technology for more than a decade with total global PV capacity reaching 303 GW at the end of 2017. Most of this capacity was met by Grid-Connected Photovoltaic (GCPV) systems.
A typical GCPV system mainly comprises PV array and inverters. A PV array consists of parallel of PV strings while each string consists of several PV modules that are connected in series. The smallest unit on a PV module is known as solar cells. These cells convert sunlight into direct current (DC) electricity. The DC power from the PV array is then converted into alternating current (AC) power using an inverter. In addition, the AC power is conditioned such that the electrical characteristics are similar to the characteristics of the grid electricity. The AC power from the inverter is later exported to the grid or consumed by the load demand.
The rapid growth of GCPV systems was expedited mainly due to the improved competitiveness, escalating electricity demand, better awareness of the potential of solar PV and the motivation of many nations in reducing GHG emissions. In addition, various incentives offered by the government had driven the solar PV market worldwide. In Malaysia, Feed-In-Tariff scheme was introduced in 2011 as part of the country's Renewable Energy Act 2011. This scheme allows owners of GCPV systems to sell the generated solar electricity to the distributed licensee by exporting the solar power to the grid. As a result, each unit of energy exported to the grid provides an income to the respective owners. In addition, since the cost of electricity generation produced by solar PV is still relatively higher than the cost of electricity produced by the conventional electricity generation, the output performance of the PV system needs to be optimized to justify its presence in the electricity generation mix.
In optimizing the system performance, the AC power output from the system as well as the weather inputs need to be monitored continuously such that the system performance for a specific monitoring period can be quantified at a later stage. Although these parameters can be monitored easily using the available data logger and sensors in the market, the AC power output from a system is usually difficult to predict as different systems are installed with different system conditions and fluctuating weather conditions [7, 8]. Most of the predic-

tion tasks are conducted using mathematical approach which requires accurate information not only about the weather conditions but also about the conditions of systems in operation. Absence of such information will dampen the accuracy of the AC power prediction.

This study presents a Multi-Layer Feedforward Neural Network (MLFNN) for predicting the output power from a GCPV system. This method is preferred when compared to the conventional mathematical approaches as no prior information about system specifications and conditions are required for the prediction.

Although the MLFNN models had shown promising prediction performance, the MLFNN models were developed based on classical design which utilizes a series of heuristic technique in selecting the training parameters using back-propagation algorithm [9]. In spite of its wide usage, the conventional heuristic-based selection of the number of neurons in hidden layer, the learning rate, the momentum rate in classical MLFNN design could be very tedious and time consuming. If the training parameters are not optimal, the network is not able to do the prediction task correctly [10]. Therefore, several guidelines were proposed to assist the selection of the ANN training parameters.

Firstly, a step-by-step procedure for selecting training parameters for MLFNN was outlined by Cooke et al. in [11]. The recommended first step towards the MLFNN training was to heuristically select the inputs and outputs, the number of neurons in hidden layer, the type of activation function and the type of learning algorithm. Then, the potential solutions of the MLFNN model based on the suitable activation functions were saved for the next stage. Then, the number of neurons in hidden layer for the saved MLFNN models was investigated. At this stage, the additional number of neurons should be restricted during the search of the minimum number of neurons such that the optimal results could be achieved. After that, the learning rate and the momentum rate were heuristically selected. Later, the required minimum number of iterative updates for each MLFNN model to converge was determined based on a minimum error set for the prediction. The training process for each model was repeated by heuristically adjusting the minimum error target such that the lowest error was achieved. Later, the trained ANN models underwent testing process. The model that produced the lowest error was selected as the optimal MLFNN model. Nevertheless, the heuristic selection of the optimal parameters of the model can be very tedious as the designers were required to manually adjust the values of training parameters and repeatedly train the network in search of the optimal model. In fact, the optimal MLFNN model might not be achieved as certain training parameters were given higher priority compared to the others.

Other than that, Laosiritaworn et al. in [12] used the design of experiment (DoE) to design the MLFNN, by referring to the statistical method to explained the correlation among the factors that give impact to the output of the network. Using this method, the optimal values for training parameters could be found. The number of training experiments was formulated based on the different combinations of all factor levels recognized for the investigation of the optimum MLFNN settings. The main factors which can affect the accuracy for the prediction task are the number of neurons in the first and second hidden layer, learning rate and momentum rate value. These factors, which are known as the factor levels, were implemented from the previous work done. Next, different MLFNN settings were derived based on the different identified factor levels. After that, the MLFNN was trained with different sets of setting and the analysis of variance (ANOVA) was conducted based on unknown testing data at 95% confidence level.

2. Methodology

An MLFNN was developed to predict the output power of a GCPV system, located at Green Energy Research Centre (GERC), Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia. There are 5 parallel strings with 3 PV modules per string connected to form a 900 Wp PV array. This PV array is then connected to a 1,200 W inverter. The inverter converts the DC power from the PV array into AC power that matches the electrical characteristics of the grid.

The MLFNN used in this study is illustrated in Figure 1. It comprises six neurons at the input layer representing six inputs and one neuron at the output layer representing one output. The inputs are current solar irradiance, SI(t), previous five-minute solar irradiance, SI(t-5), current module temperature, MT(t), previous five-minute module temperature, MT(t-5), current ambient temperature, AT(t) and previous five-minute ambient temperature, AT(t-5) while the output is current AC power from the GCPV system, $P_{ac}(t)$.

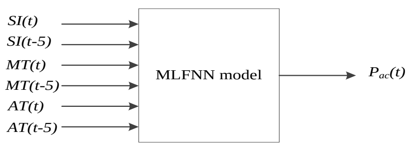

Fig. 1: Inputs and output of the MLFNN model

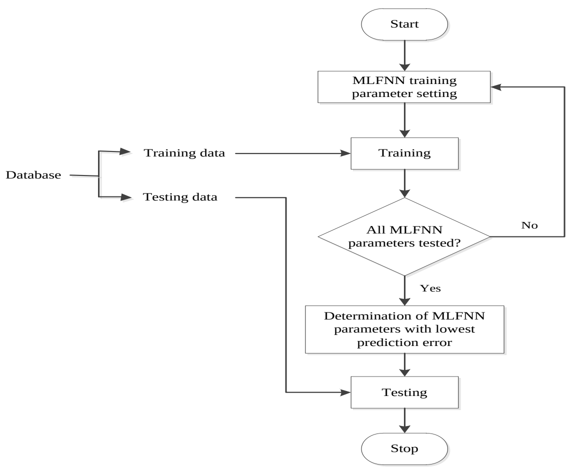
Start
MLFNN training parameter setting
Database
Training data
Testing data
Training
All MLFNN parameters tested?
No
Yes
Determination of MLFNN parameters with lowest prediction error
Testing
Stop

Fig. 2: Flowchart of classical Artificial Neural Network

The development of MLFNN was performed using Matlab software. It is conceptualized in Figure 2. Firstly, the training parameters of the network were set. Then, training was performed by simulating the network using training data. Subsequently, the prediction error was quantified and the trained network was saved. If the network had not been tested with all training parameters under study, the training process was repeated using a different set of training parameters. Otherwise, training was stopped and the best network with the lowest prediction error was determined. After that, testing was conducted by running the trained network with the lowest prediction error using testing data. Lastly, the prediction error for the testing process was determined.
In this study, the design of MLFNN involved the selection of the architecture, number of neurons in hidden layer, number of iterative updates, learning rate, momentum rate, type of activation function and type of learning algorithm.

2.1. Selection of architecture

ANN is a computational modeling tool inspired by human biological nervous system that has been used extensively for modeling complex real-world problems. Its attractiveness comes from its remarkable information processing characteristics mainly due to nonlinearity, high parallelism, fault and noise tolerance, learning and generalization capabilities [13]. Its structure comprises densely interconnected adaptive simple processing elements, known as artificial neurons or nodes. The neurons are capable of performing massively parallel computation for data processing and knowledge representation (Hecht-Nielsen, 1990; Schalkoff, 1997) [13].

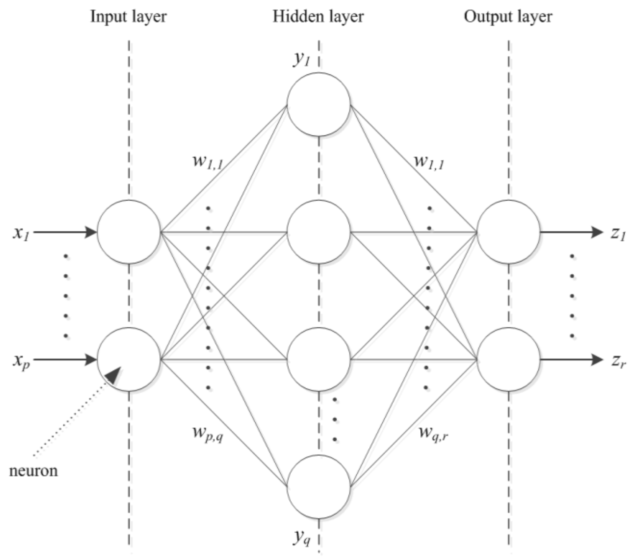

Fig. 3: A Multi-Layer Feedforward Neural Network with single hidden layer

The design of the Artificial Neural Network usually starts with the selection of ANN architecture. Although there are many ANN architectures available, the Multi-Layer Feedforward Neural Network (MLFNN) was adopted in this study. An MLFNN with single hidden layer shown in Figure 3 was selected as it has been found to be successfully used in many prediction tasks [14]. Each neuron in the input layer accepts an independent input variable whereas the neuron at the output layer produces the dependent variable. For feed-forward networks, signals are propagated from the input layer to the hidden layer and finally to the output layer. Besides that, each neuron in a layer is connected in the forward direction to each neuron in the next layer. The input of a neuron in the hidden layer, the output of a neuron in the hidden layer, and the output of a neuron in the output layer of the MLFNN are denoted as x_a, y_b and z_c respectively. The number of neurons in input layer, hidden layer and output layer are designated as p, q and r respectively. y_b is calculated using

$$y_b = f_y\left(\sum_{a=1}^{p}\left(x_a w_{a,b}\right) - B_y\right) \qquad (3.1)$$

(1)

where B_y and f_y are the bias and activation function for the neurons in the hidden layer respectively. Eventually, z_c is computed using

$$z_c = f_z\left(\sum_{b=1}^{q}\left(y_b w_{b,c}\right) - B_z\right) \qquad (3.2)$$

(2)

where B_z and f_z are the bias and activation function for the neurons in the output layer respectively.

Apart from that, the data are passed from the input layer to the output layer via several processing units. The output of a neuron in a layer is strictly dependent on the input values to that neuron and the connection weight values in the network of neurons. These values are modeled as fitting parameters for the MLFNN. The processing units, also known as the activation functions, can be linear and non-linear. Nevertheless, the activation function for every neuron in similar layer has to be of the same type. In addition, despite having similar number of inputs, the set of connection weights to each neuron in a layer can be different. Therefore, since the topology of neurons is identical, the learning process only exists in the sets of weights and biases throughout the network. A systematic algorithm to update these weights and biases is known as learning algorithm.

2.2. Selection of number of neurons

In this study, an MLFNN with single hidden layer was chosen since it is often found to be effective in many prediction tasks [15]. On the other hand, the selection of the number of neurons in the hidden layer was conducted using trial-and-error method as there is no rigid approach for the selection [15]. Nevertheless, the MLFNN cannot be trained effectively should the number of neurons in the hidden layer is very small. In contrast, if the selected number of neurons in hidden layer is very large, the training duration could be prolonged unnecessarily. In addition, too many neurons may cause over-generalization of the prediction [15]. In this study, the selection of number of neurons was performed using trial-and-error method when developing classical MLFNN models. At this stage, the number of neurons in hidden layer was varied from 1 neuron to 20 neurons before determining the best number of neurons. However, optimal selection of number of neurons was conducted using meta-heuristics when developing the hybrid MLFNN models.

2.3. Selection of learning rate and momentum rate

The learning in MLFNN was conducted by repeatedly feeding the data patterns to the network. One complete cycle of feeding these input-output data patterns is known as an iterative update, or epoch. The overall training was implemented by repeating the learning via numerous iterative updates until the optimal connection weights were obtained and the mean square error of the prediction converges to a minimum target value. On the other hand, the momentum rate value is commonly associated to the step size in weight space. The momentum rate is used to expedite the overall training process by escalating the effective step size in shallow regions of the error dimension [15]. Nonetheless, too high or too low value of momentum rate may cause slower or poor network convergence as more iterative updates are required for the convergence. Therefore, the optimal selection of momentum rate would help to accelerate the convergence. In this study, the selection of learning rate and momentum rate was performed using trial-and-error method when developing classical MLFNN models. On the contrary, optimal selection of these values was conducted using meta-heuristics when developing the hybrid MLFNN models.

2.4. Selection of activation function

The type of activation function is a linear or non-linear activation function utilized to correlate the input and the output of a neuron. In this study, the types of activation functions considered for the hidden layer were log-sigmoid (LOGSIG), hyperbolic tangent sigmoid (TANSIG), triangular basis (TRIBAS) and purely linear (PURELIN). On the other hand, PURELIN was selected as the type of activation function in the output layer.
LOGSIG takes the input with any value between plus and minus infinity and transcribes the output into the range from 0 to 1. LOGSIG was implemented using

$$f_{LOGSIG}(x) = \frac{1}{1+e^{-x}} \tag{3.3}$$

(3)

where x is the input to the neuron.

TANSIG is related to a bipolar sigmoid which has an output in the range of -1 to +1. This function is a good compromise when speed is more important than the exact shape of the activation function [16]. TANSIG was determined using

$$f_{TANSIG}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.4}$$

(4)

Another type of activation function being tested was the TRIBAS function. TRIBAS calculates a layer's output from its net input. It was computed using

$$f_{TRIBAS}(x) = 1 - |x|, -1 \leq x \leq 1 \tag{3.5}$$

(5)

Unlike the non-linear input-output characteristics shown in LOGSIG, TANSIG and TRIBAS, PURELIN mimics the linear input-output characteristics [16]. It was determined using

$$f_{PURELIN}(x) = x \tag{3.6}$$

(6)

2.5. Selection of learning algorithm

Although the weights and biases are set randomly during training process, these parameters must be changed systematically such that the error between the targeted output and the actual output from the network is consistently minimized. The rule for updating these weights and biases is commonly known as learning algorithm. The most common example of learning algorithms are Levenberg-Marquardt back-propagation (TRAINLM), Scaled Conjugate Gradient back-propagation (TRAINSCG), BFGS quasi-Newton back-propagation (TRAINBFG) and Resilient back-propagation (TRAINRP). These learning algorithms were found to be faster algorithms compared to other algorithms [17]. In this study, these algorithms were tested for all classical MLFNN models.

2.6. Selection of number of iterative updates

The number of iterative updates was determined heuristically during training of MLFNN. The number of iterative updates was determined by increasing the iterative updates from 100 to 1,000 with an increment of 100.

3. Results and discussion

The MLFNN was developed by sequentially selecting the number of hidden layer, number of iterative updates, learning rate and momentum rate, type of activation function and the type of learning algorithm. Eventually, the performance of the MLFNN model was compared with the performance of the existing model obtained from a previous work. The RMSE and computation time were used as performance indicators for the comparative studies.

3.1. Dependency on number of neurons in hidden Layer

The performance of the MLFNN model is dependent on the number of neurons as shown in Figure 4. When the number of neurons was varied from 1 to 20, the ANN with 9 neurons in hidden layer produces lowest RMSE of 24.40 W while the ANN with 2 neurons in hidden layer produces highest RMSE of 42.80 W. Thus, the best number of neurons in hidden layer is set to 9.

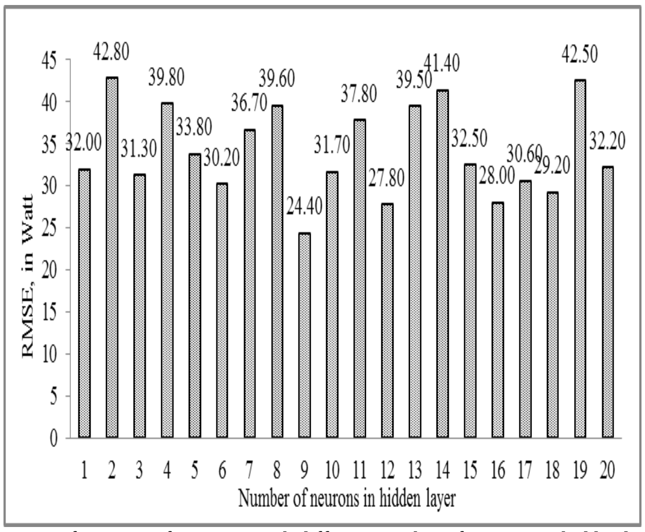

Fig. 4: Performance of MLFNN with different number of neurons in hidden layer

3.2. Dependency on number of iterative updates

Next, after determining the number of neurons in hidden layer, the selection of the best number of iterative updates for the model was conducted. The number of iterative updates was increased from 100 to 1000 with an increment of 100 iterative updates. The performance of ANN model 6 at different number of iterative updates is shown in Figure 5. The results show that lowest RMSE of 23.99 W was achieved using 1000 iterative updates while the highest RMSE of 45.24 W was obtained when the number of iterative updates is 400. Therefore, the best number of iterative updates is set to be 1000.

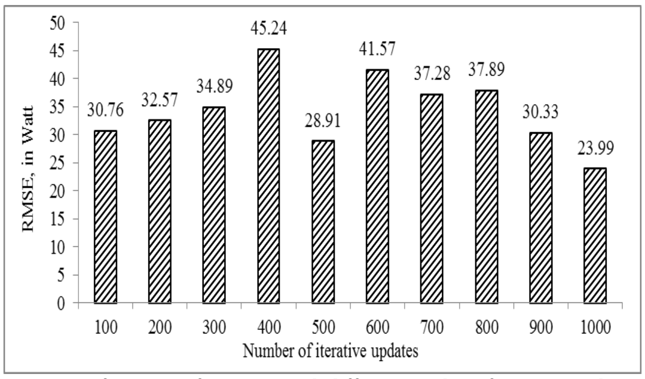

Fig. 5: Performance of MLFNN with different number of iterative updates

3.3. Dependency on learning rate and momentum rate

When the number of neurons and the iterative updates had been obtained, the set of learning rate and momentum rate were determined. The RMSE value obtained using different sets of learning rate and momentum rate value is shown in Figure 6. The best learning rate and momentum rate value were found to be 0.45 and 0.95 respectively as these values yielded lowest RMSE of 14.25 W. Meanwhile, the highest

RMSE of 42.77 W was obtained with the learning rate and momentum rate of 0.40 and 0.20 respectively. Therefore, the best learning rate and momentum rate are set to be 0.45 is 0.95 respectively.

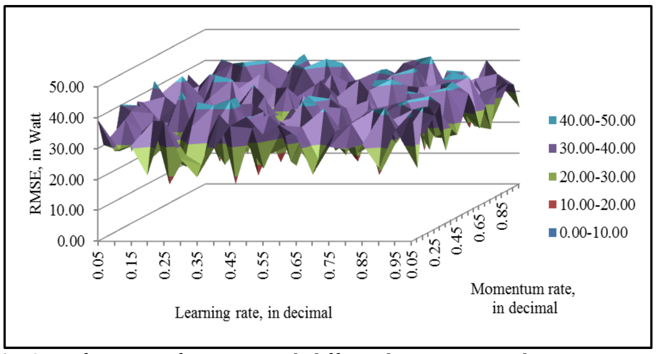

Fig. 6: Performance of MLFNN with different learning rate and momentum rate

3.4. Dependency on type of activation function

Next, the type of activation function for the hidden layer had been determined for the model. The performance of the model with four types of activation function is shown in Figure 7. The results show that LOGSIG had produced lowest RMSE of 35.72 W while TRIBAS had produced highest RMSE of 49.71 W. As a result, LOGSIG was preferred as the best type of activation function.

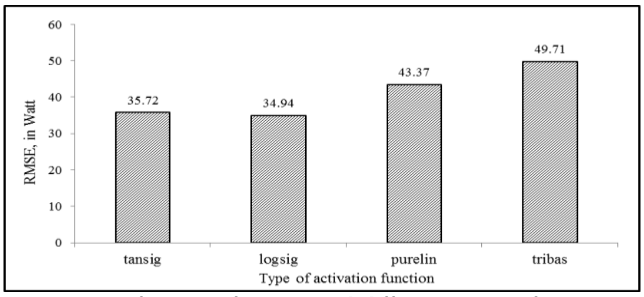

Fig. 7: Performance of MLFNN with different activation function

3.5. Dependency on type of learning algorithm

Finally, the type of learning algorithm for the model was determined. The performance of the model with selected learning algorithms is shown in Figure 8. The model with TRAINLM had exhibited lowest RMSE of 35.89 W, followed by TRAINRP, TRAINBFG and TRAINSCG. Therefore, the best type of learning algorithm for this model is selected to be TRAINLM.

3.6. Benchmarking of prediction performance

The performance of the developed MLFNN model was tested with existing MLFNN model which utilized merely SI and AT as the inputs [18]. The performance of these models is summarized in Table 1. The results showed that the developed MLFNN model is the better model as it produces the lowest RMSE of 25.6513 W during training and lowest RMSE of 32.4579 W during testing. Likewise, the model also produces highest R^2 of 0.9930 during training and 0.9844 during testing. In addition, the model also utilizes lowest computation time of 117.3636 seconds. As a result, the developed MLFNN model is not only the most accurate model but also the fastest model to be implemented in the power prediction task.

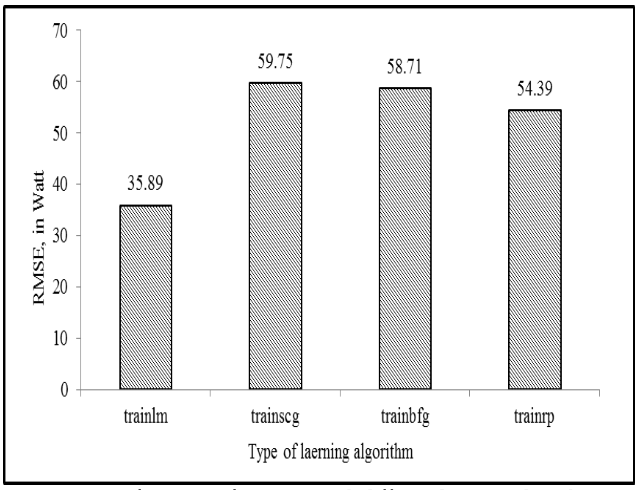

Fig. 8: Performance of MLFNN with different learning algorithm

Table 1: Final topology for prediction of AC power output using classical MLFNN

Descriptions/ ANN model	Existing	Developed
No. of neurons in input layer	2	6
No. of neurons in hidden layer	9	9
No. of neurons in output layer	1	1
No. of iterative update	700	1000
Target mean square error	10^{-2}	10^{-2}
Learning rate value	0.85	0.45
Momentum rate value	0.65	0.95
Activation function in hidden layer	LOGSIG	LOGSIG
Activation function in output layer	PURELIN	PURELIN
Learning algorithm	TRAINLM	TRAINLM
RMSE training, in Watt	35.9742	25.6513
R^2 training	0.9707	0.9930
RMSE testing, in Watt	65.0653	32.4579
R^2 testing	0.9750	0.9844
Time, in seconds	131.9647	117.3636

4. Conclusion

The study had demonstrated the usage of MLFNN as a significant tool for predicting the AC power from a GCPV system. It was showed that by incorporating MLFNN inputs with both past values and current values, the prediction accuracy had been improved. The results also showed that the inclusion of module temperature as the input to the MLFNN had improved the prediction performance.

Acknowledgement

The authors would like to express their gratitude to Green Energy Research Centre (GERC), Universiti Teknologi MARA, Shah Alam, Malaysia for making the data available for this study.

References

[1] N. Apergis, J. E. Payne, K. Menyah, and Y. Wolde-Rufael, "On the causal dynamics between emissions, nuclear energy, renewable energy, and economic growth," *Ecological Economics*, vol. 69, pp. 2255-2260, 2010.
[2] C. De Boer and I. Catsburg, "A report: The impact of nuclear accidents on attitudes toward nuclear energy," *The Public Opinion Quarterly*, vol. 52, pp. 254-261, 1988.
[3] S. Jacobsson and A. Johnson, "The diffusion of renewable energy technology: an analytical framework and key issues for research," *Energy policy*, vol. 28, pp. 625-640, 2000.
[4] A. V. Herzog, T. E. Lipman, and D. M. Kammen, "Renewable energy sources," *Encyclopedia of Life Support Systems (EOLSS). Forerunner Volume-'Perspectives and Overview of Life Support Systems and Sustainable Development,* 2001.
[5] I. Dincer, "Renewable energy and sustainable development: a crucial review," *Renewable and Sustainable Energy Reviews,* vol. 4, pp. 157-175, 2000.
[6] J. L. Sawin, F. Sverrisson, K. Seyboth, R. Adib, H. E. Murdock, C. Lins, I. Edwards, M. Hullin, L. H. Nguyen, and S. S. Prillianto, "Renewables 2017 Global Status Report," 2013.
[7] P.N.A.M. Yunus, S.I. Sulaiman and A.M. Omar, "Online performance monitoring of grid-connected photovoltaic system using hybrid improved fast evolutionary programming and artificial neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 8, no. 2, pp. 399-406, 2017.
[8] T. N. Hussain, S. I. Sulaiman, I. Musirin, S. Shaari, and H. Zainuddin, "A hybrid artificial neural network for grid-connected photovoltaic system output prediction," in *Computers & Informatics (ISCI), 2013 IEEE Symposium on*, 2013, pp. 108-111.
[9] M. H. B. KARAMI, "Application of neural networks in short-term load forecasting," *network,* vol. 1, p. 2, 2005.
[10] N. Suraweera and D. Ranasinghe, "Adaptive Structural Optimisation of Neural Networks," *ICTer,* vol. 1, 2008.
[11] M. J. Cooke and G. L. Lebby, "An optimal design method for multilayer feedforward networks," in *System Theory, 1998. Proceedings of the Thirtieth Southeastern Symposium on*, 1998, pp. 507-511.
[12] W. Laosiritaworn and N. Chotchaithanakorn, "Artificial neural networks parameters optimization with design of experiments: An application in ferromagnetic materials modeling," *Chiang Mai J. Sci,* vol. 36, pp. 83-91, 2009.
[13] I. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of Microbiological Methods,* vol. 43, pp. 3-31, 2000.
[14] D. Gao, P. Wand, and H. Liang, "Optimization of hidden nodes and training times in ANN-QSAR model," *Environ Informat Arch,* vol. 5, pp. 464-468, 2007.
[15] N. Nordin, S.I. Sulaiman and A.M. Omar, "Prediction of AC power output in grid-connected photovoltaic system using artificial neural network with multi-variable inputs," in *2016 IEEE Conference on Systems, Process and Control*, Melaka, 16-18 December 2016, pp. 192-195.
[16] M. Dorofki, A. H. Elshafie, O. Jaafar, O. A. Karim, and S. Mastura, "Comparison of artificial neural network transfer functions abilities to simulate extreme runoff data," *International Proceedings of Chemical, Biological and Environmental Engineering,* vol. 33, pp. 39-44, 2012.
[17] Ö. A. Dombaycı and M. Gölcü, "Daily means ambient temperature prediction using artificial neural network method: A case study of Turkey," *Renewable energy,* vol. 34, pp. 1158-1161, 2009.
[18] S. I. Sulaiman, I. Musirin and T. K. A. Rahman, "Prediction of grid-photovoltaic system output using three-variate ANN models," *WSEAS Transactions on Information Science and Applications,* vol. 6, no. 8, pp. 1339-1348, August 2009.