

# Prediction of Grid-Photovoltaic System Output Using Three-variate ANN Models

<sup>1</sup>SHAHIRIL IRWAN SULAIMAN, <sup>2</sup>ISMAIL MUSIRIN,

<sup>3</sup>TITIK KHAWA ABDUL RAHMAN

Faculty of Electrical Engineering  
Universiti Teknologi MARA Malaysia  
40450 Shah Alam, Selangor  
MALAYSIA

<sup>1</sup>shahril\_irwan2004@yahoo.com, <sup>2</sup>i\_musirin@yahoo.co.uk, <sup>3</sup>takitik@streamyx.com

**Abstract:** - This paper presents the prediction of total AC power output from a grid-photovoltaic system using three-variate artificial neural network (ANN) models. In this study, two-hidden layer feedforward ANN models for the prediction of total AC power output from a grid-connected photovoltaic system have been considered. Three different models were configured based on different sets of ANN inputs. In addition, each model utilizes three types of inputs for the prediction. The first model utilizes solar radiation, wind speed and ambient temperature as its inputs while the second model uses solar radiation, wind speed and module temperature as its inputs. The third model uses solar radiation, ambient temperature and module temperature as its inputs. Nevertheless, all the three models employ similar type of output which is the total AC power produced from the grid-connected system. Data filtering process has been introduced to select the quality data patterns for training process, making only the informative features are available. Thus, the regression analysis and root mean square error (*RMSE*) performance of each model could be enhanced. After the training process is completed, the testing process is performed to decide whether the training process should be repeated or stopped. Besides selecting the best prediction model, this study also exhibits some of the experimental results which illustrate the effectiveness of the data filtering in predicting the total AC power output from a grid-connected system. Each ANN model was tested with Levenberg-Marquardt training algorithm and scaled conjugate gradient training algorithm to select the best training algorithm for each model. Fully trained ANN model should later be able to predict the AC power output from a set of un-seen data patterns.

**Key-Words:** - Artificial Neural Network (ANN), photovoltaic (PV), regression coefficient (*R*), root mean square (*RMSE*), prediction, solar radiation (*SR*), ambient temperature (*AT*), wind speed (*WS*), AC power.

## 1 Introduction

Photovoltaic (PV) is a process of converting solar energy into electricity and it is widely known as solar power generation. As solar energy is available in abundance most of the daytime, solar power is deemed to be a reliable source of renewable energy technology compared to wind power, hydro power and biomass. Solar power is highly dependent on the amount of solar radiation absorbed by the solar cells on a PV module. On the other hand, the output of solar power is DC power. However, if the output power produced from a PV system is observed from the inverter output, the output is then measured in AC power.

Although PV system is able to operate as a stand-alone system, it is frequently combined with the conventional grid electricity. By having other sources of energy back-up, the reliability of the overall electricity generation system is enhanced. A grid connected PV system is becoming one of the most popular type of hybrid electricity generation involving solar power as the system only requires the PV system to be connected to the

readily available grid network provided by the local power utility. Thus, the grid could provide a source of power back-up if the PV system fails to meet the load demand. However, if the PV performance is commonly unreliable, the prime advantage of having a hybrid system becomes unachievable.

The poor performance of a PV system is related to the unpredictability of environmental conditions and PV system parameters. Firstly, the variations in sun position and changing climatic conditions may cause the PV array to produce electricity less than the load demand [1]. As the apparent motion of the sun is dissimilar throughout the year, the total irradiation received at a particular site is different from time to time. Secondly, presence of clouds and rain would decrease the irradiation received at a site due to scattering and absorption [2]. Moreover, the performance of PV array is also affected by the ambient temperature and solar cell temperature [3]. As unpredictability has become a major issue in PV system operation, ANN has been utilized to predict PV system parameters. ANN has been demonstrated to be successful

in prediction of PV system parameters as it does not require any prior information of the internal parameters of a system and conventionally involves minimal calculations [4]. For instance, load forecasting studies were conducted using ANN [5-7]. In addition, same method was employed to predict electrical energy consumption [8]. Besides predicting the load demand, ANN was also used in predicting the amount of solar radiation received at a site [9]. Furthermore, ANN was utilized to determine the size of PV system parameters with minimum input information [10-11].

This paper presents the prediction of total AC power output from a grid-photovoltaic system using three-variate ANN models. The prediction of the AC power output could be very useful for the grid-connected PV users as it could provide a good indicator of their electricity bill savings if the climatic data is known. In this study, four-layer feedforward ANN model for the prediction of total AC power output from a grid-photovoltaic system has been considered, which producing promising results.

## 2 Development of ANN Models

In the last few decades, ANN has been used extensively in various fields of power engineering for solving many complex problems. In basic computational model of ANN, a node in an ANN collects input signals from other nodes and merge them. Relevant computation is performed before the result is mapped to an output node [13]. In this study, although many ANN architectures and training algorithms have been introduced for predicting purposes, the Multi-layer Feedforward Neural Network (MLFFNN) with Levenberg-Marquardt backpropagation training algorithm has been employed due to its good generalization capability and simplicity [14]. A four-layer feedforward ANN with two hidden layers has been used to satisfactorily predict the total AC power output from a grid-connected PV system using different ANN models. An example of four-layer feedforward ANN is illustrated in Fig. 1.

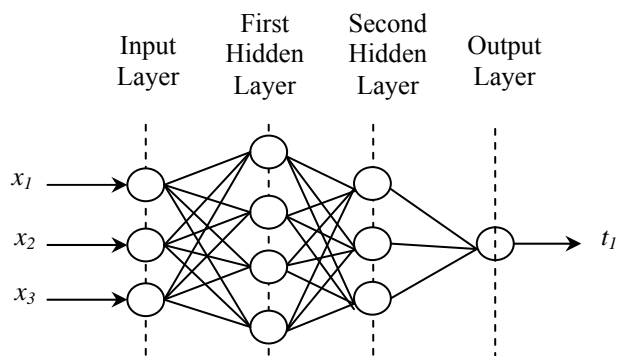


Fig. 1: An example of four-layer feedforward ANN with three inputs and one output

The value of  $x_1$ ,  $x_2$  and  $x_3$  represents the inputs of the ANN while the value of  $t_1$  represents the output of the ANN. A four-layer feedforward ANN contains an input layer, first hidden layer, second hidden layer and an output layer. In Fig. 1, the input layer comprises three neurons due to the availability of the three inputs. In this example, the first hidden layer contains 4 neurons while the second hidden layer contains 3 neurons. The number of neuron at the output layer is 1 since the ANN utilizes a single type of output. Each neuron is connected to other neurons through adaptable synaptic weights. The training of the ANN involves the adjustment of the weights such that the ANN is able to yield the desired response from a given sets of input. During training process, the weights and biases of the network are randomized and an input-output pair is chosen from the set of training data. The selected input-output pair is applied to the ANN for calculating the actual output. Later, the difference between the actual output and the targeted output is calculated to determine the error of the prediction. The selection of learning algorithm is important to minimize this error by adjusting the weights and biases of the network according to the specific learning rules used by the learning algorithm. This training process is repeated using all input-output pairs in the training data set. This training process is repeated until the mean square error goal is achieved. The overall training process will stop if the maximum allowable number of iterative updates has been reached. The training process is usually followed by the testing process to validate the prediction effort.

In this study, the data collection involves the collection of five data types including solar radiation,  $SR$  (in  $\text{kW/m}^2$ ) falling on horizontal plane, wind speed,  $WS$  (in  $\text{m/s}$ ), ambient temperature,  $AT$  (in  $^\circ\text{C}$ ), module temperature,  $MT$  (in  $^\circ\text{C}$ ) and total AC power output (in  $\text{kW}$ ). All data have been obtained from a  $42\text{kW}_p$  grid-connected PV system mounted on the roof of Quadrangle Building, University of New South Wales, Australia. The data patterns obtained are based on 15-minute interval. In this study, 1000 data patterns have been selected for each training and testing process using random partitioning.

In general, three different three-variate ANN models have been developed to predict the total AC power output from the grid-connected system. As the performance of a PV system is highly contributed by the amount of solar radiation,  $SR$  absorbed by the solar cells embedded in the PV modules,  $SR$  has become a core input type for each developed ANN model.

### 2.1 ANN Model 1 Configuration

A three-variate ANN model that uses  $SR$ ,  $WS$  and  $AT$  as its input and total AC power as its output is employed. This work is valid as performance of PV modules is also influenced by ambient temperature [12]. On the other

hand, wind speed is expected to cool down the PV array and influence the overall output performance of the grid-connected PV system. *MT* is omitted from the input configuration. The proposed ANN Model 1 is illustrated in Fig. 2.

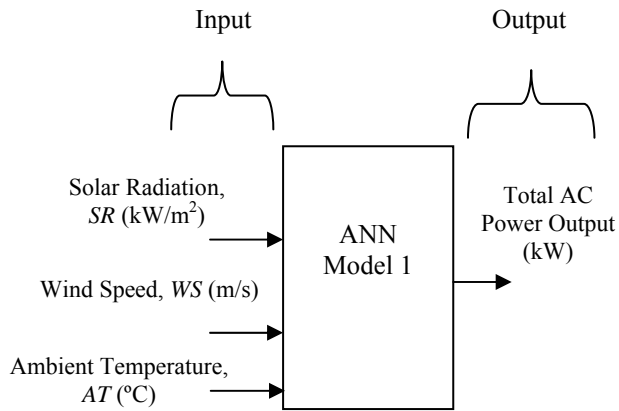


Fig. 2: ANN Model 1 based on *SR*, *WS* and *AT* as inputs

The matrix for this model is given by the following equations.

Input feature:

$$\lambda = \begin{bmatrix} x_{SR,1} & x_{SR,2} \dots & x_{SR,n} \\ x_{WS,1} & x_{WS,2} \dots & x_{WS,n} \\ x_{AT,1} & x_{AT,2} \dots & x_{AT,n} \end{bmatrix}_{3 \times n} = [\lambda_1 \quad \lambda_2 \dots \lambda_n] \quad (1)$$

where

$$\lambda_n = \begin{bmatrix} x_{SR,n} \\ x_{WS,n} \\ x_{AT,n} \end{bmatrix}$$

$n$  = number of training or testing patterns

Targeted output feature:

$$\Gamma = [t_1 \quad t_2 \dots t_n] \quad (2)$$

where  $t_n$  is the targeted output for input  $\lambda_n$ . These outputs are continuous which were assigned for the AC power output.

### 2.2 ANN Model 2 Configuration

As the PV module temperature indirectly affects the performance of solar cells in electricity conversion, the second three-variate model incorporates *SR*, *WS* and *MT* as its inputs and total AC power as its output. However, *AT* is omitted from the input configuration. The proposed ANN Model 2 is shown in Fig. 3.

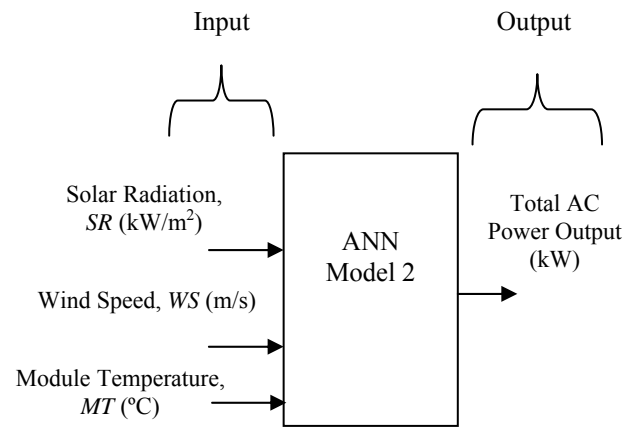


Fig. 3: ANN Model 2 based on *SR*, *WS* and *MT* as inputs

The matrix for this model is given by the following equations.

Input feature:

$$\lambda = \begin{bmatrix} x_{SR,1} & x_{SR,2} \dots & x_{SR,n} \\ x_{WS,1} & x_{WS,2} \dots & x_{WS,n} \\ x_{MT,1} & x_{MT,2} \dots & x_{MT,n} \end{bmatrix}_{3 \times n} = [\lambda_1 \quad \lambda_2 \dots \lambda_n] \quad (3)$$

where

$$\lambda_n = \begin{bmatrix} x_{SR,n} \\ x_{WS,n} \\ x_{MT,n} \end{bmatrix}$$

$n$  = number of training or testing patterns

Targeted output feature:

$$\Gamma = [t_1 \quad t_2 \dots t_n] \quad (4)$$

where  $t_n$  is the targeted output for input  $\lambda_n$ . These outputs are continuous which were assigned for the AC power output.

### 2.3 ANN Model 3 Configuration

The third model is also a three-variate model containing *SR*, *AT* and *MT* as input parameters while total AC power is used as its output. *WS* is omitted from the input configuration. This ANN model is illustrated in Fig. 4. The matrix for this model is given by the following equations.

Input feature:

$$\lambda = \begin{bmatrix} x_{SR,1} & x_{SR,2} \dots & x_{SR,n} \\ x_{AT,1} & x_{AT,2} \dots & x_{AT,n} \\ x_{MT,1} & x_{MT,2} \dots & x_{MT,n} \end{bmatrix}_{3 \times n} = [\lambda_1 \quad \lambda_2 \dots \lambda_n] \quad (5)$$

where

$$\lambda_n = \begin{bmatrix} x_{SR,n} \\ x_{AT,n} \\ x_{MT,n} \end{bmatrix}$$

$n$  = number of training or testing patterns

Targeted output feature:

$$\Gamma = [t_1 \quad t_2 \dots t_n] \quad (6)$$

where  $t_n$  is the targeted output for input  $\lambda_n$ . These outputs are continuous which were assigned for the AC power output.

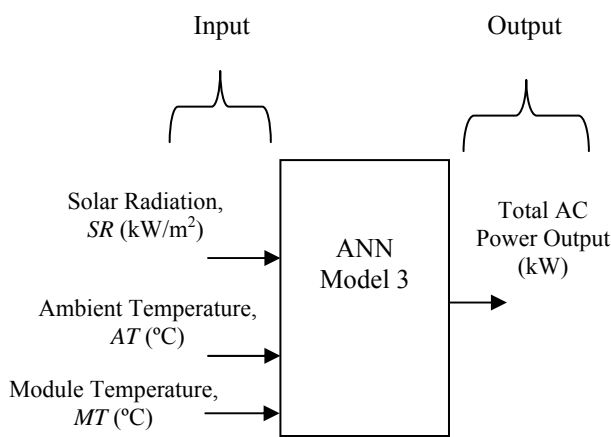


Fig. 4: ANN Model 3 based on  $SR$ ,  $AT$  and  $MT$  as inputs

### 3 Development of Prediction Program

In this study, three ANN models have been developed. After developing the three ANN models based on available type of inputs, a prediction program utilizing two-hidden layer feedforward ANN is developed. Data normalization has been conducted in making sure that the range is in between -1 and +1. The process was based on equation (1).

$$p_{norm} = N_{min} + \left[ (N_{max} - N_{min}) \times \left[ \frac{p_{actual} - R_{min}}{R_{max} - R_{min}} \right] \right] \quad (7)$$

where  $p_{norm}$  is the normalized value for a particular data and  $p_{actual}$  is the actual value of a particular data.  $R_{min}$  is the minimum actual value of data selected from the available data patterns of the same type whereas  $R_{max}$  is the maximum actual value of data selected from the available data patterns of the same type.  $N_{max}$  and  $N_{min}$  represent the normalized range for the normalize data values. The values of  $N_{max}$  and  $N_{min}$  have been chosen to be 1 and -1 respectively.

During the training process, extensive experiments have been conducted to determine the optimal number of

nodes in each hidden layer, momentum rate, learning rate and other training parameters. The number of neurons in the first hidden layer, the number of neurons in the second hidden layer, the type of activation functions, number of iterative updates (epochs) and the mean square error goal were varied to search for the optimal types or values. After extensive investigation, the logarithmic-sigmoid activation function is selected at each hidden layer to model the non-linear relationship between the ANN inputs. However, Levenberg-Marquardt algorithm (*trainlm*) and scaled conjugate gradient algorithm (*trainscg*) are tested for each ANN model due to their good track records in predicting PV system outputs [15-18]. Prediction performance of the ANN is quantified by computing the regression coefficient,  $R$  [dimensionless], absolute error,  $E_{abs}$  [in kW] and Root Mean Square Error,  $RMSE$  [in kW]. While the value of  $R$  is determined using the neural network toolbox in MATLAB, the  $E_{abs}$  and  $RMSE$  are calculated as follows:

$$E_{abs} = |a - t| \quad (8)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (a_i - t_i)^2}{n}} \quad (9)$$

where  $a$  is the actual forecasted value of output and  $t$  is the target value of output.  $n$  is the number of training or testing patterns involved. The regression analysis is used to minimize the prediction error when fitting the predicted output to the targeted output [19]. The best architecture of ANN Model 1, Model 2 and Model 3 are selected separately based on the lowest  $RMSE$  and highest  $R$  obtained. The search of this model is further accelerated by the usage of data filtering section built in the program. This section is designed to eliminate training patterns which produces  $E_{abs}$  values of greater than or equal to 2kW. In addition, this preset value is estimated based on a maximum of 5% error of the maximum potential power to be generated by the 42kW PV system.

The testing process is conducted consecutively after obtaining the best trained ANN model. Each model is also trained twice using the two different training algorithms mentioned previously. The overall procedure for the prediction program using ANN can be summarized as follows:

- i. Start training process by loading training data.
- ii. Adjust ANN architecture and training parameters.
- iii. Perform training process.
- iv. If the training converges, proceed to the next step. Otherwise, return to step ii.
- v. Determine absolute error,  $E_{abs}$  for each training pattern, regression coefficient,  $R$  and Root Mean Square Error,  $RMSE$ .
- vi. If  $R$  of training is greater or equal to 0.99,

- vii. proceed to the next step. If not, return to step ii.
- viii. If the prediction produces patterns with  $E_{abs}$  greater or equal to 2kW, proceed to the next step. Otherwise, go to step ix.
- ix. Perform data filtering. Remove data patterns which produce  $E_{abs}$  greater or equal to 2kW. Then, return to step iii.
- x. Save trained ANN.
- xi. Start testing process by loading testing data and recalling the trained ANN in training process.
- xii. Perform testing process using the successfully trained ANN.
- xiii. If the testing converges, proceed to the next step. Otherwise, return to step ii.
- xiv. Determine the regression coefficient,  $R$  and Root Mean Square Error,  $RMSE$  for testing.
- xv. If  $R$  is greater than or equal to 0.99, the testing process is stopped. Otherwise, return to step ii.

After determining the best ANN architecture and training parameters for each ANN model, the best prediction model for this study is chosen based on the model that produces lowest  $RMSE$  and highest  $R$ . The best training algorithm is also selected for each prediction model.

## 4 Results and Discussion

After training and testing the different ANN models, the performance of each model was analyzed and compared to determine the best prediction model. The results of this work can be categorized into four sections. The first section describes the best architecture and training parameters for the three different models based on different training algorithms used, while the second section illustrates the prediction performance of each model using *trainlm*. The third section describes the prediction performance of each model using *trainscg*. Finally, based on the overall results in previous sections, the last section reveals the best model for the prediction of total AC power output from a grid-connected PV system.

### 4.1 Architecture and Training Parameters of different ANN Models

After extensive investigation, all the three models obtain the same best configuration of transfer function. The best configuration is [log-sigmoid, log-sigmoid, purely-linear]. Apart from that, the learning rate of 0.05 and momentum rate of 0.05 are found to be sufficient for all training processes using either *trainlm* or *trainscg*. The best ANN architecture and optimum training parameters for each model using *trainlm* are tabulated in Table 1. In Table 1, Despite having the same configuration of transfer function, further investigation shows that the three models require different neural configuration

depending on different sets of ANN inputs as well as the type of training algorithm employed. Using *trainlm*, Model 1 and Model 2 have similar neural configuration of (2,4,1) whereas Model 3 has a configuration of (3,2,1). These results imply that Model 1 and Model 2 require almost the same neural configuration to be realized. It is also observed that training data of Model 2 experiences the most data filtration compared to the other two models. In contrast, data for Model 3 is the least affected by data filtering.

Table 1: Results for ANN Architecture and Training Parameters before filtering using *trainlm*

| Parameters / Results                         | ANN Model 1           | ANN Model 2           | ANN Model 3           |
|--|-----------------------|-----------------------|-----------------------|
| Types of elements in input layer             | SR, WS, & AT          | SR, WS & MT           | SR, AT & MT           |
| Number of training patterns before filtering | 1000                  | 1000                  | 1000                  |
| Number of training patterns after filtering  | 859                   | 848                   | 866                   |
| Number of testing patterns                   | 1000                  | 1000                  | 1000                  |
| Number of neurons in first hidden layer      | 2                     | 2                     | 3                     |
| Number of neurons in second hidden layer     | 4                     | 4                     | 2                     |
| Number of elements in output layer           | 1                     | 1                     | 1                     |
| Type of transfer function                    | logsig-logsig-purelin | logsig-logsig-purelin | logsig-logsig-purelin |
| Number of Epochs                             | 1000                  | 1000                  | 1000                  |
| Mean Square Error Goal                       | 0.001                 | 0.001                 | 0.001                 |

On the other hand, the best neural topology for each model using *trainscg* is described in Table 2. Although the best transfer function configuration for each model is found to be [log-sigmoid, log-sigmoid, purely-linear], the optimal neural configuration is discovered to be different for each model. The optimal neural configuration for Model 1 and Model 2 are found to be (3,2,1) and (4,3,1) respectively. The optimal configuration for Model 3 is (2,3,1). Using *trainscg*, Model 2 experiences the most data filtration while Model 1 has the minimum influence from data filtration.

Before filtering process, many data points having

absolute prediction error of more than 2kW are obtained. However, after filtering process, all predicted *RMSE* values are reduced to values below 2kW. Consequently, the *RMSE* performance is expected to improve as the number of data points with  $E_{abs}$  greater than or equal to 2kW are removed from the training patterns.

Table 2: Results for ANN Architecture and Training Parameters before filtering using *trainscg*

| Parameters / Results                         | ANN Model 1                   | ANN Model 2                   | ANN Model 3                   |
|--|-------------------------------|-------------------------------|-------------------------------|
| Types of elements in input layer             | <i>SR, WS, &amp; AT</i>       | <i>SR, WS &amp; MT</i>        | <i>SR, AT &amp; MT</i>        |
| Number of training patterns before filtering | 1000                          | 1000                          | 1000                          |
| Number of training patterns after filtering  | 847                           | 825                           | 838                           |
| Number of testing patterns                   | 1000                          | 1000                          | 1000                          |
| Number of neurons in first hidden layer      | 3                             | 4                             | 2                             |
| Number of neurons in second hidden layer     | 2                             | 3                             | 3                             |
| Number of elements in output layer           | 1                             | 1                             | 1                             |
| Type of transfer function                    | logsig-<br>logsig-<br>purelin | logsig-<br>logsig-<br>purelin | logsig-<br>logsig-<br>purelin |
| Number of Epochs                             | 1000                          | 1000                          | 1000                          |
| Mean Square Error Goal                       | 0.001                         | 0.001                         | 0.001                         |

#### 4.2 Prediction Performance of ANN Models using Levenberg-Marquardt Training Algorithm

After the training and testing process were conducted consecutively for each model using different training algorithms, the prediction performance of each setting was recorded. In Table 3, using *trainlm*, Model 1 experiences the highest improvement of 0.49% in *R* value from 0.99381 to 0.99872 after filtering. On the other hand, after filtering process, Model 2 produces an increase of *R* value from 0.99383 to 0.99826 with approximately 0.45% improvement while Model 3 has received an improvement of 0.43% in *R* value from 0.99433 to 0.99858. Although Model 1 yields the lowest *R* value before filtering, the model actually obtains not only the highest improvement of *R* value after filtering but also the highest *R* value during testing. In contrast,

despite having the highest *R* value before filtering process, Model 3 produces the least improvement due to filtering. However, Model 3 produces *R* value which is slightly lower than *R* value of Model 1 during testing process. These results show that the data filtering has the highest influence on Model 1. In addition, Model 1 can also be considered as having the best *R* performance when *trainlm* is employed during training process mainly due to the highest *R* value obtained during training and testing.

Table 3: Regression Performance for Each ANN Model Using *trainlm*

| Performance Type                                    | ANN Model 1 | ANN Model 2 | ANN Model 3 |
|---|-------------|-------------|-------------|
| <i>R</i> before filtering for training              | 0.99381     | 0.99383     | 0.99433     |
| <i>R</i> after filtering for training               | 0.99872     | 0.99826     | 0.99858     |
| Percentage of <i>R</i> improvement for training (%) | 0.49        | 0.45        | 0.43        |
| <i>R</i> for testing                                | 0.99367     | 0.99274     | 0.99351     |

Table 4: Root Mean Square Error Performance for Each ANN Model Using *trainlm*

| Performance Type                                     | ANN Model 1 | ANN Model 2 | ANN Model 3 |
|--|-------------|-------------|-------------|
| <i>RMSE</i> before filtering for training (kW)       | 1.2103      | 1.1842      | 1.1358      |
| <i>RMSE</i> after filtering for training (kW)        | 0.5434      | 0.6368      | 0.5729      |
| Percentage of <i>RMSE</i> reduction for training (%) | 55.10       | 46.23       | 49.56       |
| <i>RMSE</i> for testing (kW)                         | 1.3649      | 1.4747      | 1.3971      |

Apart from that, the *RMSE* performance of each model using *trainlm* is tabulated in Table 4. Before filtering process, Model 3 has the lowest *RMSE* value whereas Model 1 has the highest *RMSE* value. Nevertheless, after filtering process, Model 1 has the highest improvement of *RMSE* value with 55.10% reduction of *RMSE* from 1.2103kW to 0.5434kW. After filtering process, Model 1 also has the lowest *RMSE* among the three ANN models. The lowest *RMSE* reduction is 46.23% which is attained by Model 2. The *RMSE* of Model 2 is decreased from 1.1842kW to 0.6368kW. Despite having the lowest *RMSE* before filtering, Model 3 only experiences a reduction of 49.56%

in *RMSE* value from 1.1358kW to 0.5729kW. In testing process, Model 1 exhibits the lowest *RMSE* value of 1.3649kW while Model 2 shows the highest *RMSE* value of 1.4747kW. Therefore, the percentage of *RMSE* reduction indicates that the filtering action is most effective on Model 1 and is least effective on Model 2. Besides that, Model 1 also produces the lowest *RMSE* value during training and testing. Thus, Model 1 has the best *RMSE* performance among the three models when *trainlm* is used. The prediction results for each model are depicted graphically in Fig. 5 to 7. All the three models indicate satisfactory prediction with most actual forecasted output data match the targeted data.

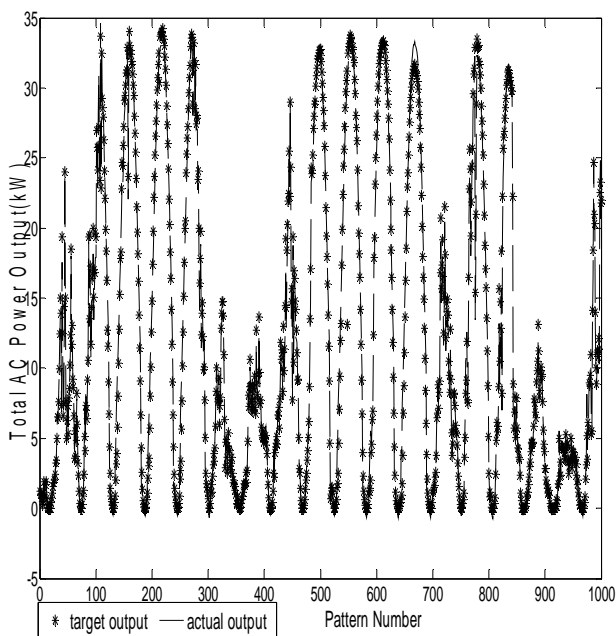


Fig. 5: Model 1- Prediction results during testing (with *trainlm*)

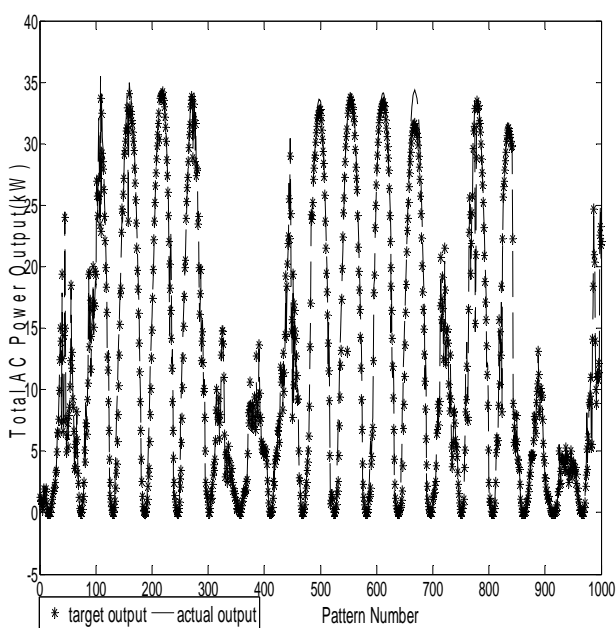


Fig. 6: Model 2- Prediction results during testing (with *trainlm*)

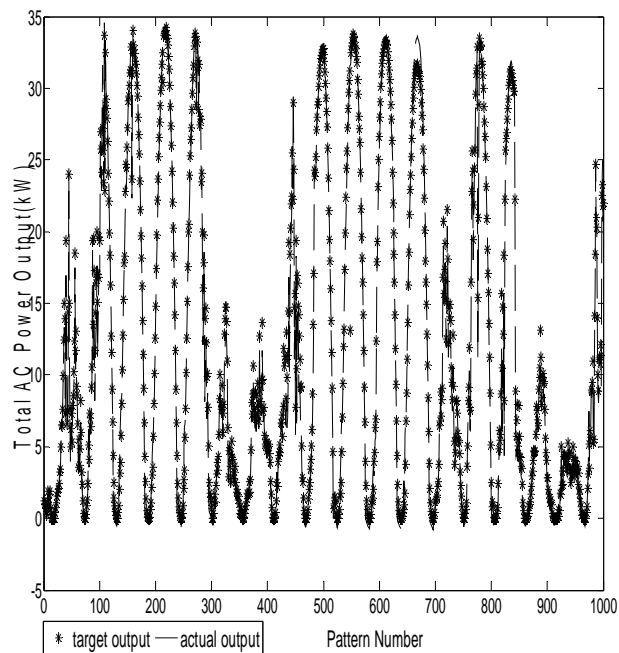


Fig. 7: Model 3- Prediction results during testing (with *trainlm*)

### 4.3 Prediction Performance of ANN Models using Scaled Conjugate Gradient Training Algorithm.

The prediction performance of each model using *trainscg* training technique is tabulated in Table 5 and Table 6. Using *trainscg*, Model 1 shows the highest improvement of *R* value among the three models. The *R* value of Model 1 increases 0.54% from 0.99331 to 0.99870. In contrast, Although Model 3 has the highest *R* value before and after filtering during training process, it still has the lowest improvement of *R* value among the three models. The *R* value of Model 3 increases 0.52% from 0.99356 to 0.99874. Model 2 shows an improvement of 0.53% of its *R* value from 0.99348 to 0.99873. However, after filtering process, it is observed that the percentage of improvement on *R* and the values of *R* obtained from the three models are almost similar. During testing, Model 2 exhibits the highest *R* value of 0.99304 while Model 3 shows the lowest *R* value of 0.99265. As the value of *R* after filtering process is almost similar for all the three models where the *R* value for Model 2 is the highest during the testing, Model 2 can be accepted as the model with the best regression performance.

In terms of *RMSE* performance as tabulated in Table 6, Model 2 shows the lowest *RMSE* of 0.5454kW after filtering although Model 3 initially exhibits the lowest *RMSE* of 1.2102kW before filtering process. Nevertheless, Model 1 still shows the highest reduction of *RMSE* value. The *RMSE* value of Model 1 has been reduced for about 55.66% from 1.2329kW to 0.5467kW. The lowest *RMSE* reduction of 54.84% is achieved by

Model 3 with *RMSE* value being reduced from 1.2101kW to 0.5465kW. Model 2 experiences an *RMSE* reduction of 55.20% from 1.2173kW to 0.5454kW. During testing process, Model 1 yields the lowest *RMSE* value of 1.3633kW while Model 2 exhibits the highest *RMSE* of 1.4310kW. As the *RMSE* value for each model is approximately the same after filtering process and the *RMSE* value of Model 1 is the lowest during testing process, Model 1 can be regarded as the best prediction model with the best *RMSE* performance when *trainscg* is utilized as the training algorithm.

Table 5: Regression Performance for Each ANN Model Using *trainscg*

| Performance Type                                    | ANN Model 1 | ANN Model 2 | ANN Model 3 |
|---|-------------|-------------|-------------|
| <i>R</i> before filtering for training              | 0.99331     | 0.99348     | 0.99356     |
| <i>R</i> after filtering for training               | 0.99870     | 0.99873     | 0.99874     |
| Percentage of <i>R</i> improvement for training (%) | 0.54        | 0.53        | 0.52        |
| <i>R</i> for testing                                | 0.99274     | 0.99304     | 0.99265     |

Table 6: Root Mean Square Error Performance for Each ANN Model Using *trainscg*

| Performance Type                                     | ANN Model 1 | ANN Model 2 | ANN Model 3 |
|--|-------------|-------------|-------------|
| <i>RMSE</i> before filtering for training (kW)       | 1.2329      | 1.2173      | 1.2102      |
| <i>RMSE</i> after filtering for training (kW)        | 0.5467      | 0.5454      | 0.5465      |
| Percentage of <i>RMSE</i> reduction for training (%) | 55.66       | 55.20       | 54.84       |
| <i>RMSE</i> for testing (kW)                         | 1.3633      | 1.4310      | 1.3665      |

The prediction results for each model during testing

process are depicted graphically in Fig. 8 to 10. All the three models show satisfactory prediction results with most actual forecasted output data being close to the targeted output. Therefore, the prediction results have been validated.

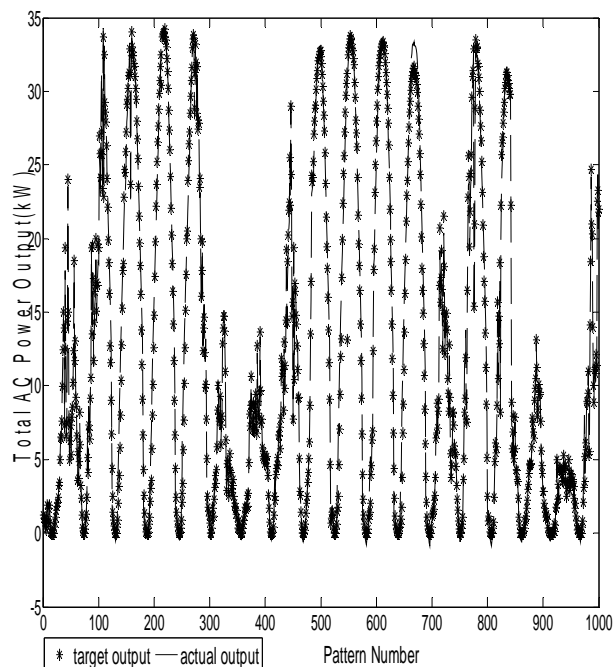


Fig. 8: Model 1- Prediction results during testing (with *trainscg*)

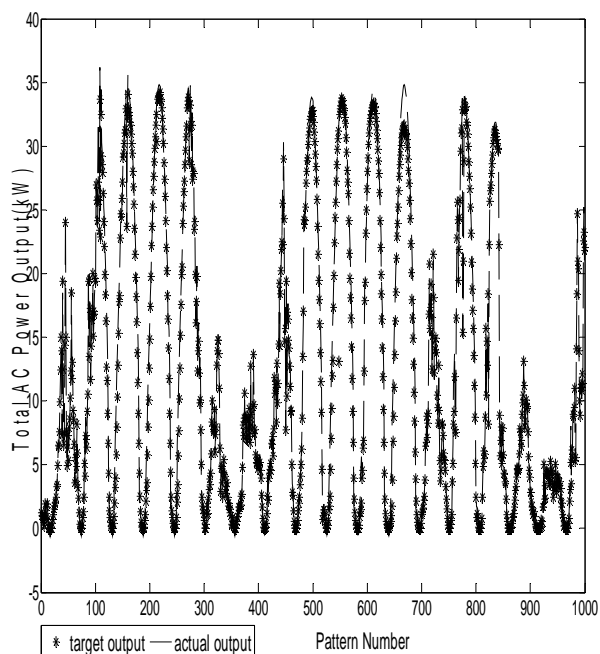


Fig. 9: Model 2- Prediction results during testing (with *trainscg*)



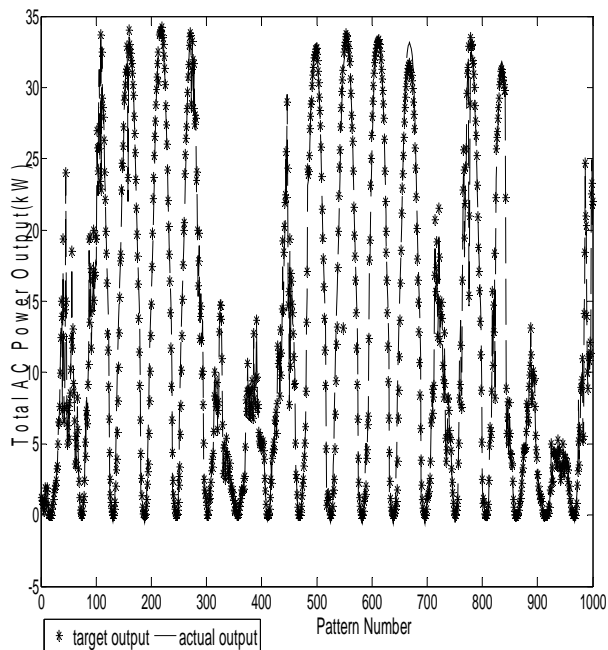


Fig. 10: Model 3- Prediction results during testing (with *trainscg*)

#### 4.4 Selection of the Best ANN Model for the Prediction Task

Based on the results from the earlier sections, ANN Model 1 has been selected as the best model for the prediction task. Using *trainlm*, the model yields the highest  $R$  value and the lowest  $RMSE$  value during testing process. On the other hand, the model also produces comparatively high value of  $R$  and lowest  $RMSE$  value using *trainscg* during testing process. Apart from that, Model 1 has the largest improvement on both  $R$  and  $RMSE$  performance after filtering action. This finding also indicates that data filtering is most suitable for Model 1. Although *trainlm* and *trainscg* are found to be significant in the training of Model 1, *trainlm* is selected as the best training algorithm for Model 1 due to the higher value produced using this technique. The  $RMSE$  performance is almost the same when both training algorithms are employed in Model 1.

## 5 Conclusion

This work has presented the prediction of total AC power output from a grid-connected photovoltaic system using three-variate ANN models. The aim of the study is to produce the best model for predicting the total AC power output from a grid-connected PV system based on three input variables. It is found that the ANN Model 1 which utilizes solar radiation, wind speed and ambient temperature as its input variables is selected as the best model for the prediction task. It is also shown that all the three models are capable of predicting the total AC power output as the  $R$  values obtained are sufficiently closed to unity, i.e. all  $R$  values are greater than 0.99. In addition,

the  $RMSE$  performance of all models using both algorithms is almost similar during the prediction task. Apart from that, *trainlm* is discovered to perform better than *trainscg* frequently. However, based on  $R$  performance, it can be observed that *trainlm* is more suitable to be implemented for Model 1 and Model 3 while *trainscg* is more suitable for Model 2. Besides that, data filtering is found to be beneficial in enhancing the performance of the prediction models although this technique might require less data patterns to be used for training. In short, the ANN models proposed by this study is expected to be useful for predicting the total AC power output of grid-connected PV system from an unknown climatic data set.

#### References:

- [1] H. Asano, K. Yajima, and Y. Kaya, Influence of Photovoltaic Power Generation on Required Capacity for Load Frequency Control, *IEEE Trans. on Energy Conversion*, Vol. 11, No. 1, pp.188-193, 1996.
- [2] S.R. Wenham, M.A. Green, and M.E. Watt. *Applied Photovoltaics*, Centre for Photovoltaic Devices and Systems, Sydney: The University of New South Wales, 1995.
- [3] S. Roberts, *Solar Electricity*, Hertfordshire: Prentice Hall International (UK), 1991.
- [4] M. Balzani and A. Reatti, Neural Network Based Model of a PV Array for the Optimum Performance of PV System, in *Proc. 2005 PhD Research in Microelectronics and Electronics Conf.*, Vol. 2, pp. 123-126.
- [5] T. Senjyu, H. Takara, K. Uezato, and T. Funabashi, One-hour-ahead Load Forecasting Using Neural Network, *IEEE Transactions on Power Systems*, Vol. 17, No. 1, 2002, pp. 113-118.
- [6] H.S Hippert, C.E. Pedreira, and R.C. Souza, Neural Networks for Short-term Load Forecasting: A Review and Evaluation, *IEEE Transactions on Power Systems*, Vol. 16, No. 1, 2001, pp. 44-55.
- [7] D.C. Park, R.J. El-Sharkawi, and R.J. Marks II, Electric Load Forecasting Using an Artificial Neural Network, *IEEE Trans. on Power Syst.*, Vol. 6, No. 2, 1999, pp. 442-449.
- [8] G.E. Nasr, E.A. Badr, and M.R. Younes, Neural Network in Forecasting Electrical Energy Consumption: Univariate and Multivariate Approaches, *Int. Journal of Energy Research*, Vol. 26, 2002, pp. 67-68.
- [9] A.S.S. Dorvlo, J.A. Jervase, and A. Al-Lawati, Solar Radiation Estimation Using Artificial Neural Networks, *Applied Energy*, Vol. 71, 2002, pp. 307-319.
- [10] A. Mellit, M. Benhanem, A. Hadj Arab, and A. Guessoum, Modelling of Sizing the Photovoltaic

- System Parameters Using Artificial Neural Network, in *Proc. IEEE Control Applications Conf.*, Vol. 1, 2003, pp.353-357.
- [11] L. Hontoria, J. Aguilera, and P. Zufiria, A Tool for Obtaining the LOLP Curves for Sizing Off-Grid Photovoltaic Systems Based in Neural Networks, *World Conf. on Photovoltaic Energy Conversion, 2003*, 11-18 May 2003, pp. 2423 – 2426.
- [12] I. Ashraf and A. Chandra, Artificial Neural Network Based Models for Forecasting Electricity Generation of Grid Connected Solar PV Power Plant, *Int. Journal of Global Energy Issues*, Vol. 21, No. 1/2, 2004, pp. 119-130.
- [13] K.L Priddy and P.E. Keller, *Artificial Neural Networks: An Introduction*, New Delhi: Prentice Hall of India, 2007.
- [14] S. Kumar, *Neural Networks A Classroom Approach*, Singapore: McGraw-Hill, 2005.
- [15] S.I. Sulaiman, T.K. Abdul Rahman, I. Musirin, and S. Shaari, Assessment of Different Training Algorithms in ANN Model for Grid-Photovoltaic System Output Prediction, *Progress of Solar Energy Research and Development*, 21-22 October 2008, pp. 101-107.
- [16] M. Hayati and Z. Mohebi, Application of Artificial Neural Networks for Temperature Forecasting, *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 22, 2007, pp. 275-279.
- [17] O.A. Dombayci and M. Golcu, Daily Means Ambient Temperature Prediction Using Artificial Neural Network Method: A Case Study of Turkey, *Renewably Energy*, Vol. 34, 2009, pp. 1158-1161.
- [18] M. Hayati and Y. Shirvany, Artificial Neural Network Approach for Short Term Load Forecasting for Illam Region, *Proceedings of World Academy of Science, Engineering and Technology*, Vol. 22, 2007, pp. 280-284.
- [19] T.W. Gentry, B.M. Wiliamowski and L.R. Weatherford, A Comparison of Traditional Forecasting Techniques and Neural Networks, *Intelligent Engineering Systems Through Artificial Neural Networks*, Vol. 5, 1995, pp. 765-770.