

Neural Network-Based Estimation of Solar Radiation Level

Joshua Kasirye¹, Lynnte Jane Nazziwa¹, Herman Wahid^{2*}

¹Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

²Process Tomography and Instrumentation Research Group, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

DOI: <https://doi.org/10.51244/IJRSI.2024.11120061>

Received: 17 December 2024; Accepted: 20 December 2024; Published: 17 January 2025

ABSTRACT

This paper presents the design and implementation of an Artificial Neural Network (ANN) model to estimate solar radiation using meteorological data for a four seasoned country, specifically in Sydney, Australia. The model aims to provide a cost-effective alternative to direct measurement by leveraging available data such as temperature, humidity, wind speed, sea level pressure and rainfall. The model's performance is evaluated using two methods: a GUI-based neural network toolbox and custom MATLAB codes. Performance metrics, including Mean Squared Error (MSE) and the correlation coefficient (R), were assessed to identify the most effective model. The results show that ANN models can accurately predict solar radiation levels, offering important insights for adaptive solar energy systems. The study concludes by comparing the two approaches, emphasizing their respective strengths and limitations. This research highlights the potential of utilizing ANNs with easily accessible meteorological data to improve the efficiency of solar energy harvesting systems.

Keywords: Artificial Neural Network (ANN), Solar Radiation Estimation, Meteorological Data, MATLAB.

INTRODUCTION

Solar radiation is a critical parameter in the design and operation of solar energy systems. Accurate estimation of solar radiation can significantly enhance the performance of solar tracking systems [1]. Traditional measurement methods involve sensors like pyranometers that can be expensive and require regular maintenance, e.g., in Fig.1. This study explores the use of Artificial Neural Networks (ANN) to estimate solar radiation using meteorological data as inputs. This can be a smart substitute to solar radiation measurement that can significantly reduce costs of calibration and sensor installation [2, 3].

An Artificial Neural Network (ANN) is a computational model inspired by the way biological neural networks in the human brain process information. ANNs consist of interconnected nodes (neurons) organized into layers. These networks learn to perform tasks by adjusting the connections (weights & biases) based on input data and output error [4, 5]. These weights and bias are also known as the model parameters. ANNs are particularly effective in modelling complex relationships in data, making them suitable for estimating solar radiation based on meteorological parameters [1, 6]. They learn more efficiently from large datasets which can be an important advantage of using ANN models [7, 8].



Fig. 1. Traditional way of Solar radiation measurement.

The paper is structured as follows: Section 2 discusses the dataset preparation. Section 3 describes the ANN model, its construction using a GUI-based toolbox. Section 4 outlines the implementation of the ANN model using MATLAB code. Section 5 compares the performance of the models, and Section 6 concludes the study with final remarks.

Dataset Preparation

Data Collection

For this study, meteorological data for one year in 2023 was collected from five stations in Sydney, Australia namely, *Randwick*, *Richmond*, *Rozelle*, *Vineyard* and *St. Mary's*, all available at *visualcrossing.com* weather data. The dataset includes daily records of minimum, maximum and average temperature, average relative humidity, dew, maximum wind speed, wind gust and wind direction, cloud cover, sea level pressure, visibility and solar radiation for every day amounting to a total of 1,825 data rows. All the data is structured and numerical in nature. The data was preprocessed by removing any non-numerical data and then divided into 70% (1,277) training, 15% (274) validation, and 15% (274) testing sets for the model development.

Data Normalization

Normalization was done to ensure that all inputs equally contribute to the model's training process and so that it can converge faster. Each feature in the dataset was scaled to a range between 0 and 1 using the min-max normalization formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X is the original feature value, X_{min} and X_{max} are the minimum and maximum values of the feature, respectively. This was done on the earlier obtained dataset by the help of python and Visual Studio Code IDE. The python code used can be found in the appendix [9-11].

Ann Model Development

MATLAB Model using MATLAB GUI

Using MATLAB's neural network toolbox (*nnstart*), several ANN models were constructed and evaluated. The following steps were undertaken:

Input Selection: Various combinations of input parameters were tested to determine the best input-output mapping. This involved selecting from the 13 features which can be used as inputs. The output was only the solar radiation across all the combinations.

Model Training: Different models based on different inputs were trained more than once using the Levenberg-Marquardt backpropagation fitting algorithm. The training that ended with the lowest converging Mean Square Error or higher regression R value was considered for that set of input parameters. The network structure is shown in the image below. With only one input layer, one hidden layer, and one output layer. The hidden layer nodes number was varied and different results obtained. Fig. 1 shows an ANN model with 13 inputs, 10 nodes in Hidden layer and 1 output layer.

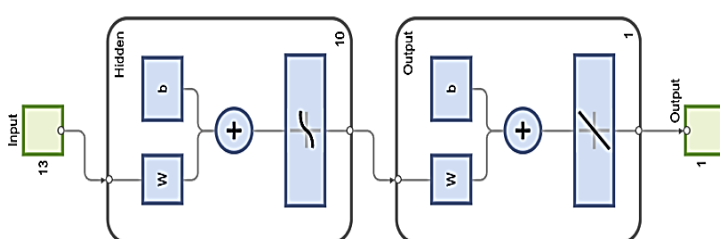


Fig. 2. The Artificial Neural Network Model Structure

Performance Evaluation: The number of hidden neurons, epochs, simulation time, and error performance were recorded. Plots for training performance, fitting, and regression were generated. From the information in these plots, the models were evaluated.

MATLAB Model using MATLAB codes

The ANN model was also constructed using MATLAB codes without the GUI-based toolbox but with MATLAB inbuilt scripts editor. The feedforward gradient descent backpropagation (*newff* and *traingdx* commands) and radial basis function (*newrb* command) methods were implemented.

The two methods show more insights in which of the two methods has a faster convergence and higher R value. Then the results are also compared with those obtained from training the model using the toolbox.

Input Data: The complete normalized dataset was imported into MATLAB as Comma Separated Value (csv) file. Then the data was separated as features and target, where target is the solar radiation data. After this stage, the data was then split into 70% training, 15 % validation and 15% testing.

Model Training: Different models were trained based on the two different methods namely *newff*, and *newrb*. The training that ended with the lowest converging Mean Square Error or higher regression R value was considered a better one.

Different models with different hidden layer nodes and epochs were trained and results are shown in the next section. The network structure is similar to that of the GUI. However, the difference is that the weight and bias update functions and techniques.

Performance Evaluation: The number of hidden neurons, epochs, simulation time, and error performance were recorded. Plots for training performance, MSE values, and regression were generated. From the information in these plots, the models were evaluated.

RESULTS AND DISCUSSION

The performance of the models constructed using both the MATLAB GUI and MATLAB code were compared based on factors such as training time, error performance (MSE and R), and the accuracy of solar radiation estimation from the testing dataset.

ANN models using MATLAB with GUI

The GUI toolbox provided generally faster response as most of the training took less than 8 second irrespective of the number of neurons in the hidden layer. The number of neurons was varied from 1 to 256. The MSE and R results are shown in Fig. 3a-3c and the MSE performance graph for 32, 64 and 256 neurons models are depicted in Fig. 4. The inputs number is also varied from 7 to 13, where the result is shown in Fig. 5.

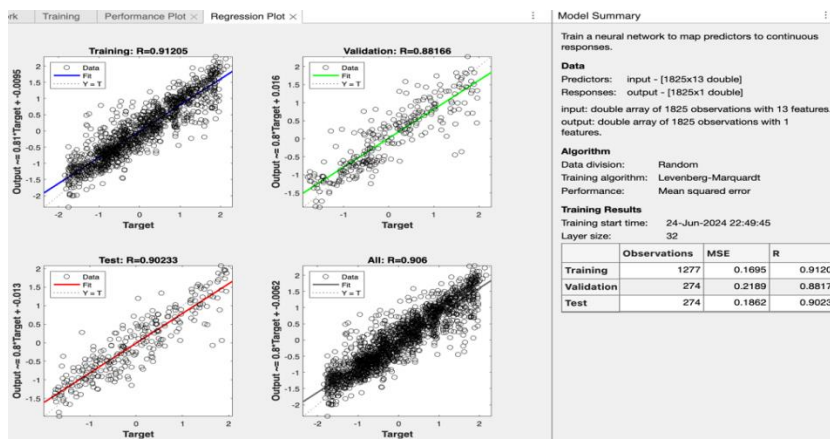


Fig. 3a. MSE and R values for 13 inputs and 32 hidden neurons.

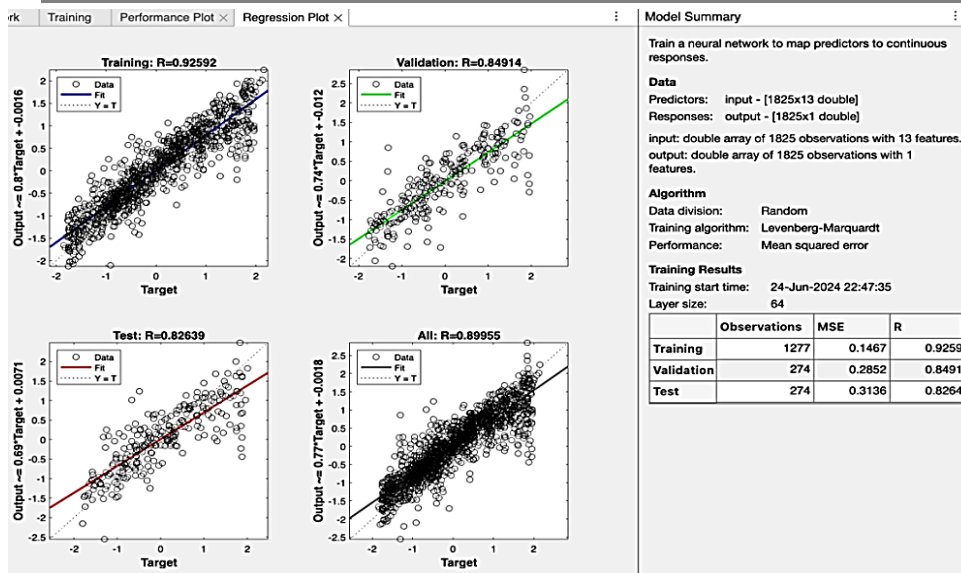


Fig. 3b. MSE and R values for 13 inputs and 64 hidden neurons.

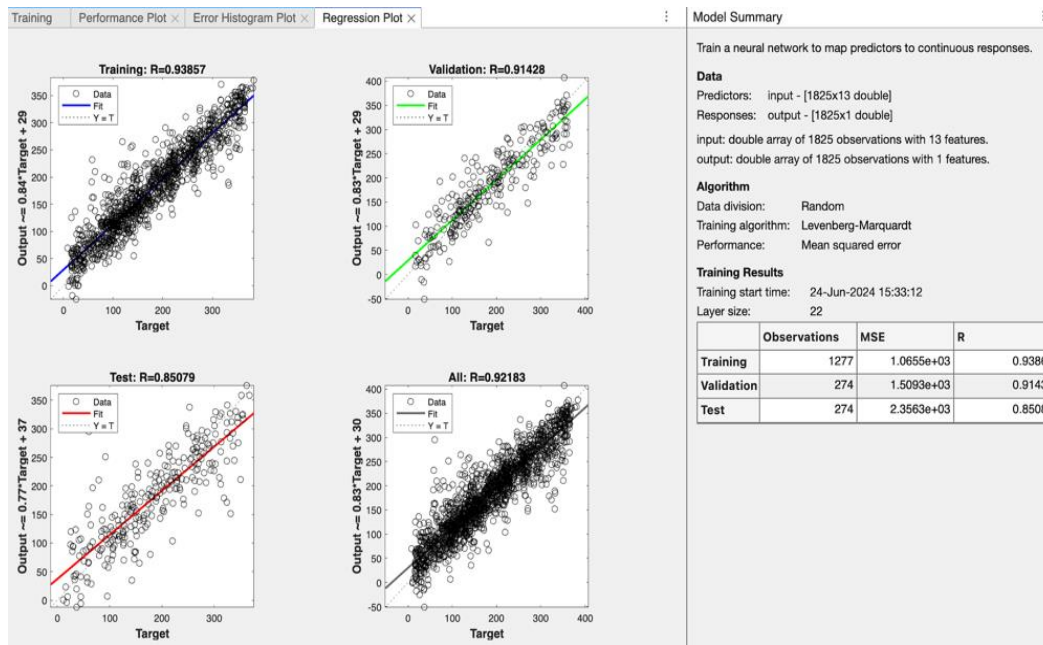


Fig. 3c. MSE and R values for 13 inputs and 256 hidden neurons.

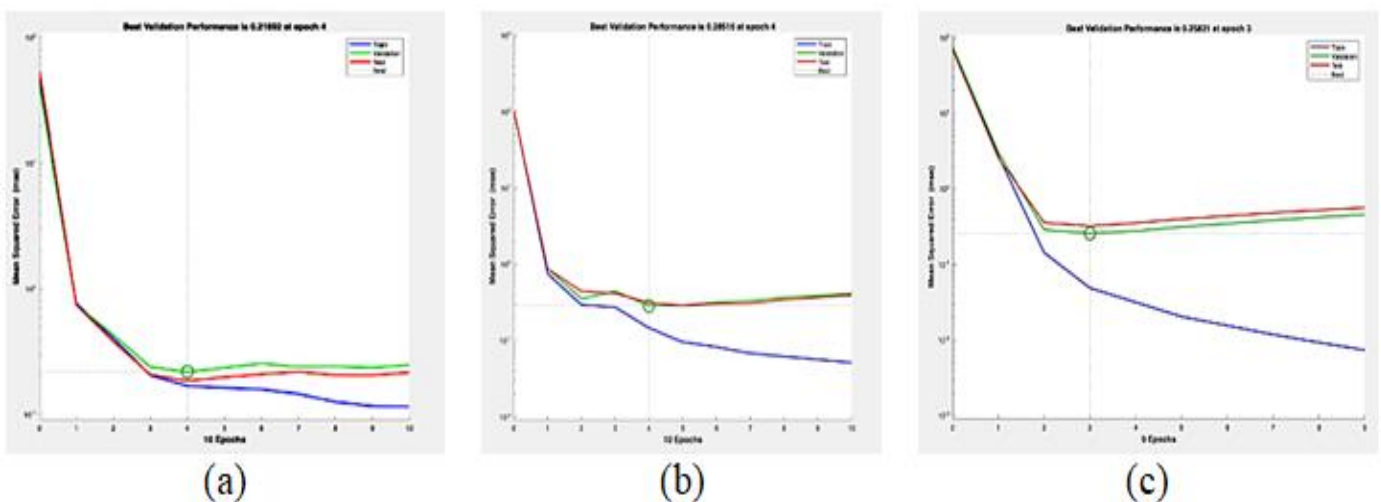


Fig. 4. MSE performance graphs for (a) 32, (b) 64 and (c) 256 neurons models.

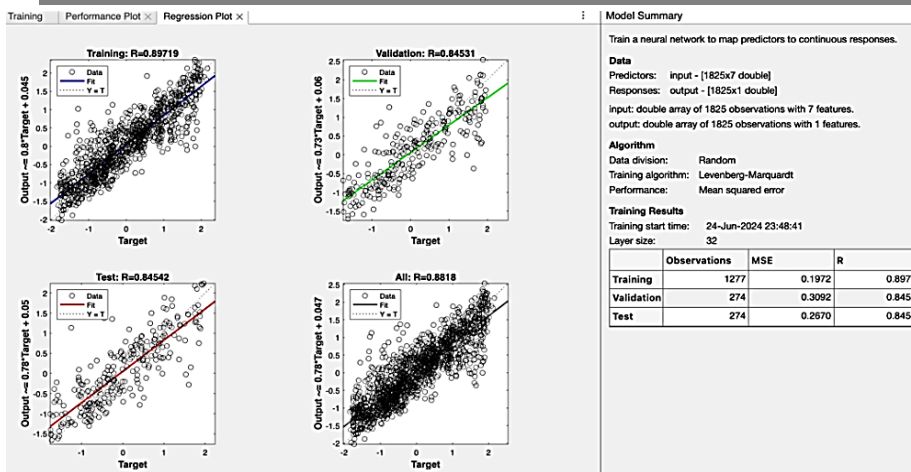


Fig. 5. MSE and R values for 7 inputs and 32 hidden neurons

For the fewer input parameters, 7 input parameters were selected which could have a significant effect on solar radiation, inputs like day, month, maximum temperature, dew, humidity, wind speed and cloud cover. Generally, having more parameters helped the model converge faster and with a less validation error.

From Fig. 3 above, it was observed that as the number of hidden layer neurons increased, the model’s training R value increased while its MSE error reduced significantly. However, this made the training accuracy an overfit as compared to the rest of the validation and testing accuracies as shown in Fig. 4.

It was also observed that as the number of input parameters, and hidden layer neurons increased, the model’s training time increased but with a lower MSE errors and more clear regression graphs. More hidden layer neurons provided better overall accuracy of the model. The summary of GUI model results is given in Table 1.

Table 1. Summary of the GUI models result.

Number of Inputs	Number of Neurons	Overall R value	Train MSE	Validation MSE	Test MSE
13	32	0.9060	0.1695	0.2189	0.1862
13	64	0.8996	0.1467	0.2852	0.3136
13	256	0.9380	0.0489	0.2582	0.3218
7	32	0.8818	0.1972	0.3092	0.2870

From Table 1 above, it was difficult to conclude on which of the models was better than the other because those with high R values, had also some high MSE errors. However, due to a very low training MSE error and highest R value and average Validation MSE values, the model with 256 neurons and all 13 inputs was chosen.

ANN models using MATLAB codes

MATLAB based ANN models provided a better step by step design of the models and this gave a better hands-on experience on what happens inside the MATLAB GUI toolbox. 13 input parameters we used mostly and once, the 7 input parameters on both methods. 256 was chosen as the maximum number of neurons in the hidden layer and other parameters were varied for better comparison.

Feedforward gradient descent (‘newff’) training method

The similar format of obtaining the results using ‘newff’ training methods are illustrated in Fig. 6 to Fig. 10. From these figures, it was observed that as the number of neurons in the hidden layer increased, the R value

was lower and very high MSE errors. This same effect was observed with even a smaller number of inputs. However, it had a fast execution time. It was deduced that this method was not as accurate as compared to the GUI method. It required very a small number of neurons hence it could be interpreted as not a functional ANN. We also had a big MSE error even with the smaller number of neurons ranging from 0.31 to 19.5. The summary of 'newff' training model results is given in Table 2.

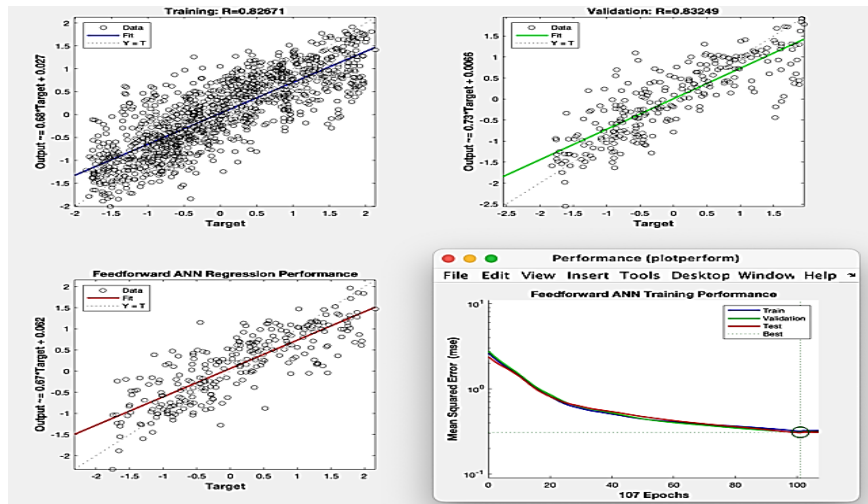


Fig. 6. The *newff* ANN model performance for 13 inputs and 9 hidden neurons.

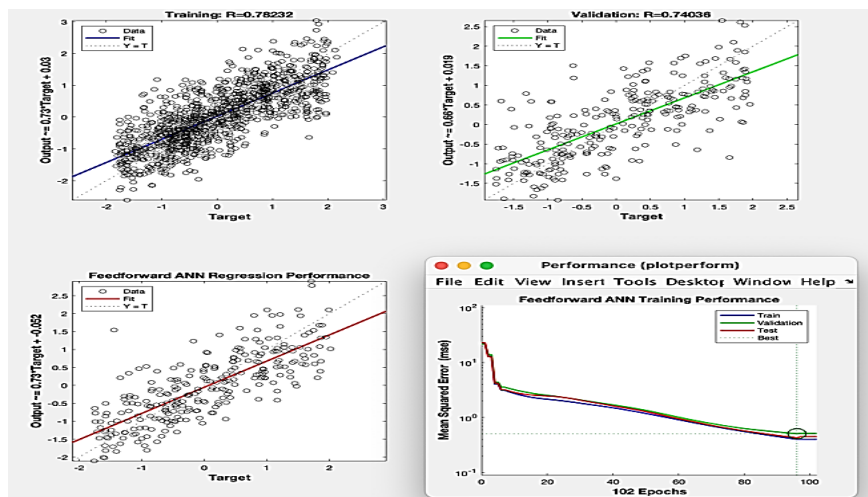


Fig. 7. The *newff* ANN model performance for 13 inputs and 64 hidden neurons.

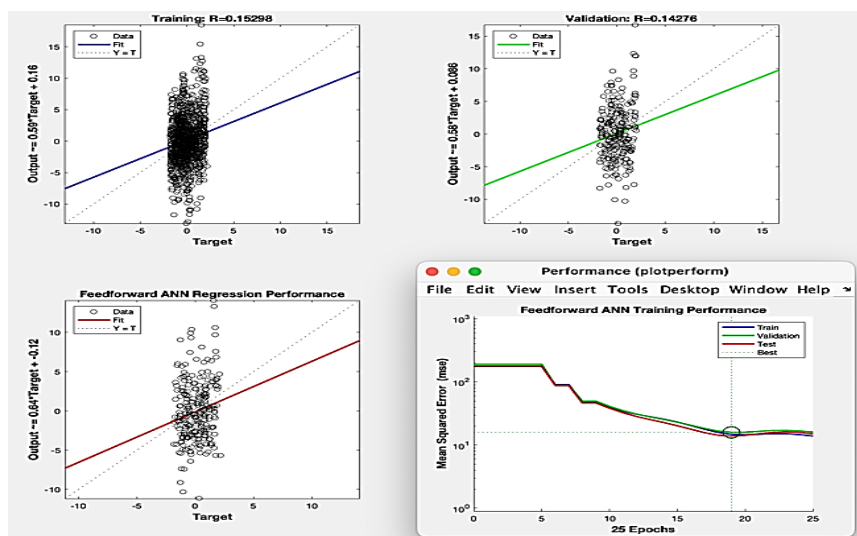


Fig. 8. The *newff* ANN model performance for 13 inputs and 256 hidden neurons.

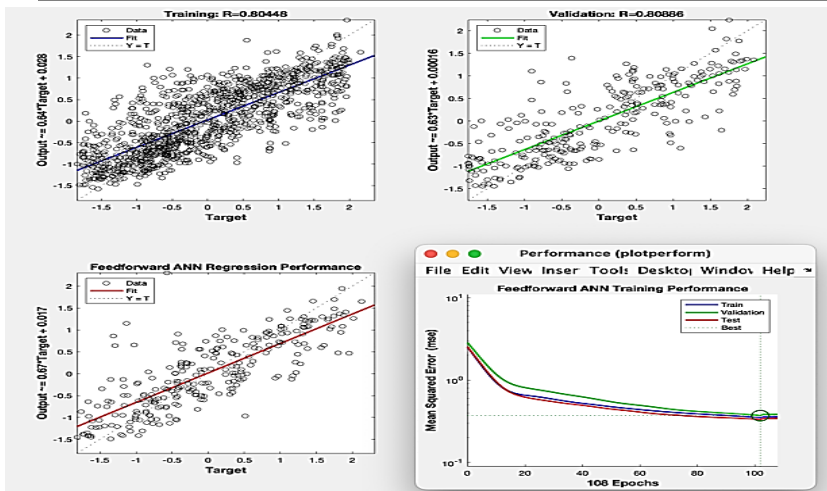


Fig. 9. The *newff* ANN model performance for 7 inputs and 9 hidden neurons.

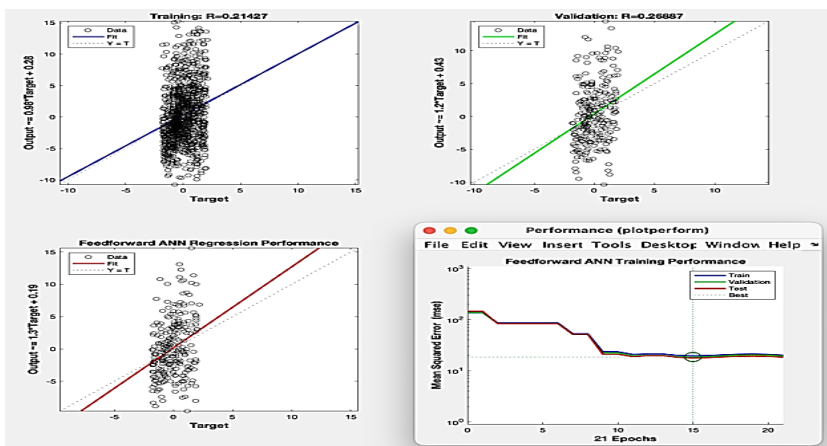


Fig. 10. The *newff* ANN model performance for 7 inputs and 256 hidden neurons.

Table 2. Summary of the *newff* codes models result.

Inputs	Neurons	R value	Train MSE	Validation MSE	Test MSE	Run time
13	9	0.8296	0.3188	0.3078	0.3045	2s
13	13	0.8122	0.3439	0.3486	0.3227	3s
13	64	0.7613	0.4034	0.5062	0.4237	4s
13	128	0.6079	0.8472	0.9852	1.0008	5s
13	256	0.1479	14.837	15.988	13.913	5s
7	9	0.8067	0.3515	0.3711	0.3329	1s
7	256	0.2416	19.534	18.434	17.722	3s

Radial basis function (*newrb*) training method

The similar format of obtaining the results using *newrb* training methods are illustrated in Fig. 11 to Fig. 15. From these figures, it was observed that as the number of neurons in the hidden layer increased, the R value was higher and very low MSE errors. This same effect was observed with even a smaller number of inputs. However, it had a low execution time of about 20 seconds. It was deduced that the *newrb* method was more

accurate than newff and the GUI methods. It exhibited the true advantage of high number of neurons in the ANN modeling. The *newrb* method exhibited the smallest MSE errors for training, validation and testing even after comparing with the GUI method. It also showed that as the number of epochs increased in the training, the easier it was for the model to converge and with a very small MSE value and high accuracy of about 95%.

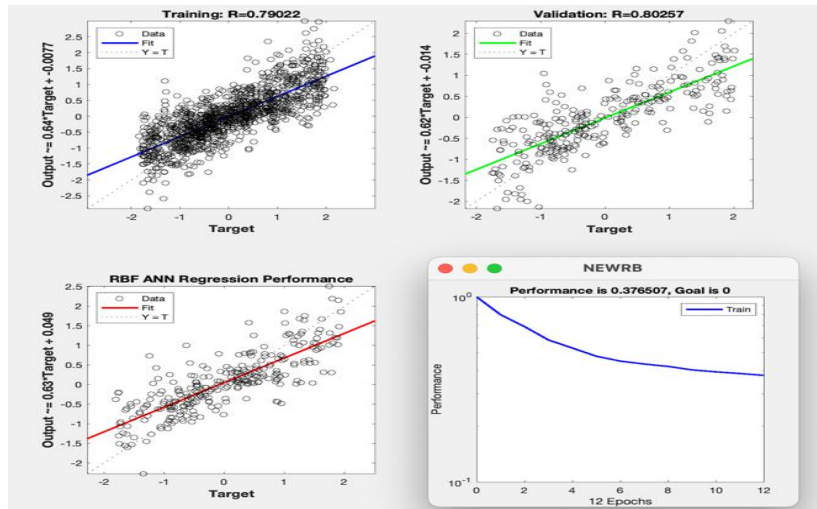


Fig. 11. The *newrb* ANN model performance for 13 inputs and 13 hidden neurons.

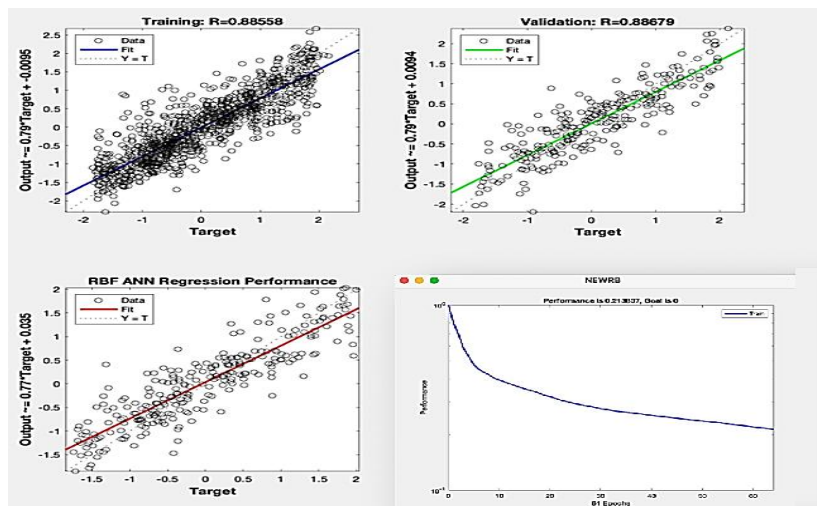


Fig. 12. The *newrb* ANN model performance for 13 inputs and 64 hidden neurons.

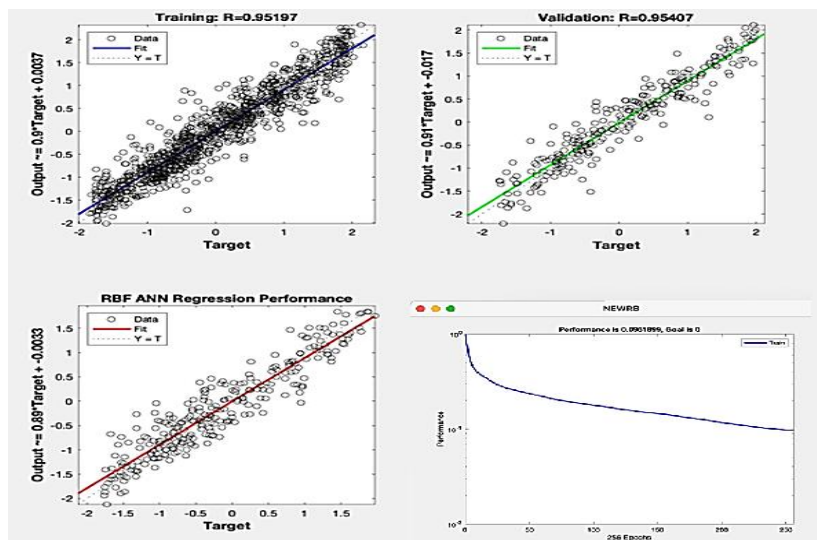


Fig. 13. The *newrb* ANN model performance for 13 inputs and 256 hidden neurons.

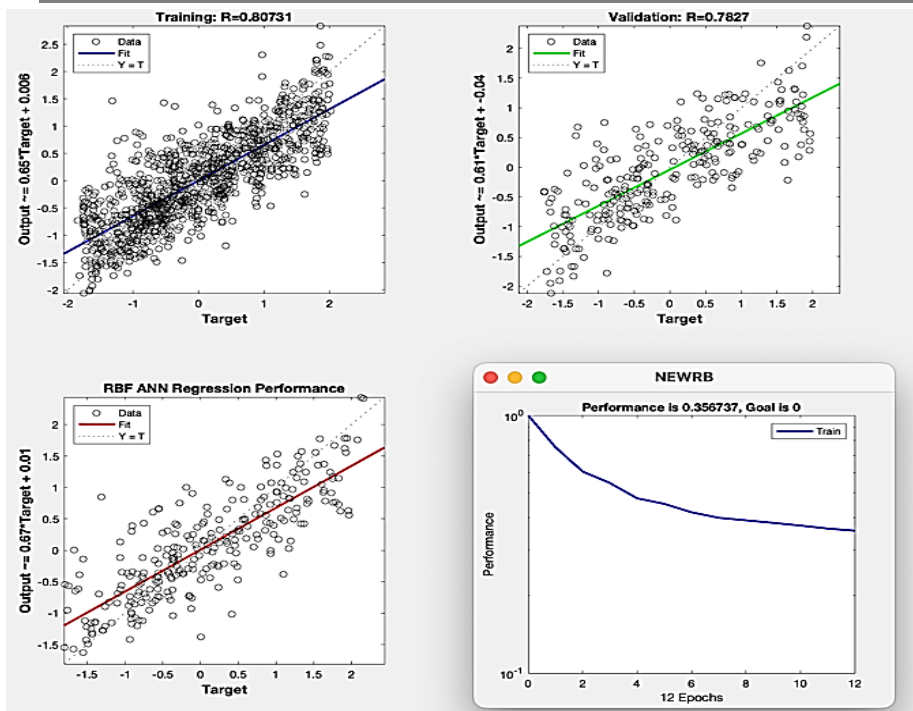


Fig. 14. The *newrb* ANN model performance for 7 inputs and 13 hidden neurons.

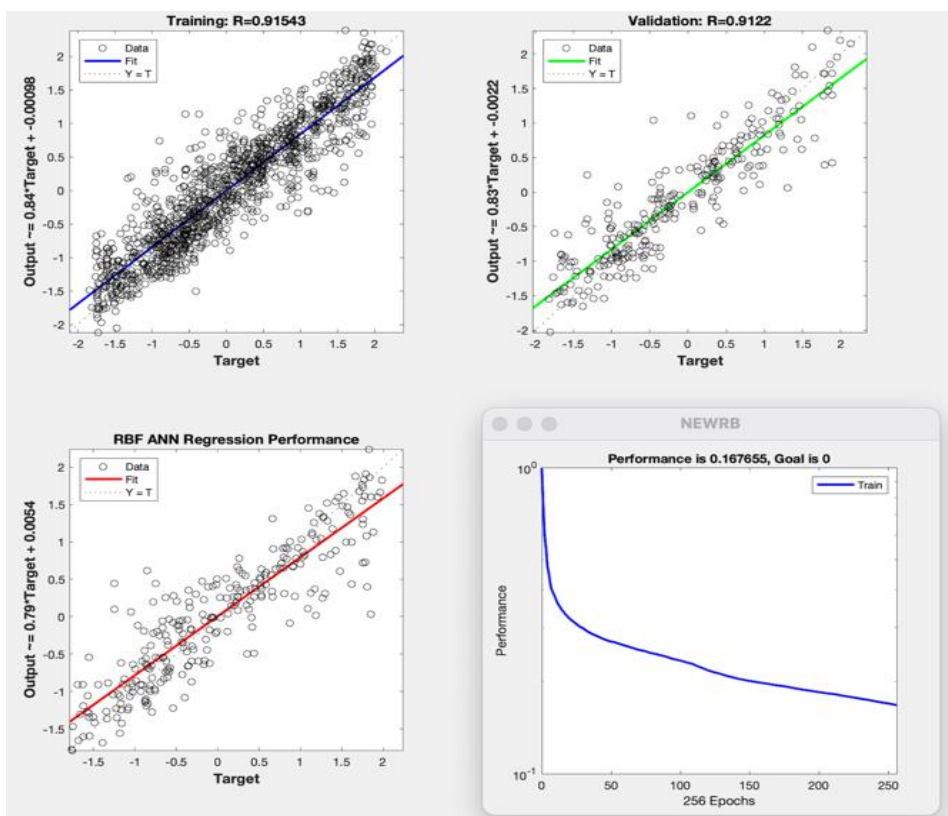


Fig. 15. The *newrb* ANN model performance for 7 inputs and 256 hidden neurons.

Table 3. Summary of the *newrb* codes models result.

Inputs	Neurons	R value	Train MSE	Validation MSE	Test MSE	Run time
13	13	0.7964	0.3713	0.3934	0.331	5s
13	64	0.8862	0.2208	0.2005	0.1949	10s

13	256	0.953	0.0952	0.0872	0.1098	20s
7	13	0.795	0.3455	0.4109	0.3194	3s
7	256	0.9138	0.1616	0.1638	0.1995	18s

The overall comparison of the models' performance is listed in Table 4. From the table, it was observed that method using MATLAB code with the help of the newrb method provided a more accurate ANN model for solar radiation measurement. Future research could involve deploying the model by using its final weights and bias for further real time application[1, 10]. From the accuracy obtained, this model is good enough for further field implementation and testing.

Table 4. Summary of the models Performance Comparison.

ANN Method Used	R value	Train MSE	Validation MSE	Test MSE	Run time
GUI toolbox	0.9380	0.0489	0.2582	0.3218	1s
Newff	0.8296	0.3188	0.3078	0.3045	2s
Newrb	0.9530	0.0952	0.0872	0.1098	20s

CONCLUSION

The performance of the models constructed using both methods (i.e. GUI-based and Matlab codes) was compared. The comparison was based on factors such as training time, error performance, and the accuracy of solar radiation estimation. There was a problem of overfitting as the number of neurons increased and yet few input parameters were used especially while using the GUI toolbox. There was a tradeoff between accuracy and runtime (computational power). Models that trained slowly had more accurate result as compared to those which trained faster. The increase in the number of neurons and epochs affected the models different but significantly. It was deduced that as the number of epochs and neurons increased, the more accurate the *newrb* model was. Whereas the reverse pattern was observed in the other two models.

REFERENCES

1. Premalatha, N., and Valan Arasu, A. Prediction of solar radiation for solar systems by using ANN models with different back propagation algorithms. *Journal of Applied Research and Technology. JART*, 10.1016/j.jart.2016.05.001 vol. 14, no. 3, pp. 206-214, doi: 10.1016/j.jart.2016.05.001 (2016).
2. Rahman, S., Rahman, S., and Bahalul Haque, A. K. M. Prediction of Solar Radiation Using Artificial Neural Network. *Journal of Physics: Conference Series*, vol. 1767, no. 1, p. 012041, 2021/02/01, doi: 10.1088/1742-6596/1767/1/012041 (2021).
3. Yadav, A., and Chandel, S. Solar radiation prediction using Artificial Neural Network techniques: A review. *Renewable and Sustainable Energy Reviews*. vol. 33, pp. 772-781, 05/01 doi: 10.1016/j.rser.2013.08.055 (2014).
4. Qazi, A., Hussain, F., Al-Fatlawi, A., Raj, R., Abd Rahim, N., and Khan, W. The artificial neural network for solar radiation prediction and designing solar systems: a systematic literature review. *Journal of Cleaner Production*. vol. 104, 04/27 doi: 10.1016/j.jclepro.2015.04.041 (2015).
5. Bączkiewicz, A., Wątróbski, J., Sałabun, W., and Kołodziejczyk, J. An ANN Model Trained on Regional Data in the Prediction of Particular Weather Conditions. *Applied Sciences*. vol. 11, no. 11, doi: 10.3390/app11114757 (2021).
6. Purnomo, H., Hartomo, K., and Prasetyo, S. Artificial Neural Network for Monthly Rainfall Rate Prediction. *IOP Conference Series: Materials Science and Engineering*. vol. 180, p. 012057, 03/01 doi: 10.1088/1757-899X/180/1/012057 (2017).
7. Ridwan, W., Sapitang, M., Aziz, A., Kushiar, K., Ali Najah Ahmed, A.-M., and El-Shafie, A., Rainfall

- forecasting model using machine learning methods: Case study Terengganu, Malaysia, *Ain Shams Engineering Journal*. vol. 12, 11/01 doi: 10.1016/j.asej.2020.09.011 (2020).
8. Faceira, J., Afonso, P., and Salgado, P. Prediction of Solar Radiation Using Artificial Neural Networks. in *CONTROLO'2014 – Proceedings of the 11th Portuguese Conference on Automatic Control*, Cham, A. P. Moreira, A. Matos, and G. Veiga, Eds., : Springer International Publishing, pp. 397-406 (2015).
 9. Marzouq, M., Fadili, H. E., Lakhliai, Z., and Zenkouar, K. A review of solar radiation prediction using artificial neural networks. in *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 19-20 April 2017, pp. 1-6, doi: 10.1109/WITS.2017.7934658 (2017).
 10. Hou, X., Ju, C., and Wang, B. Prediction of solar irradiance using convolutional neural network and attention mechanism-based long short-term memory network based on similar day analysis and an attention mechanism. *Heliyon*, vol. 9, no. 11, p. e21484, 2023/11/01/, doi: <https://doi.org/10.1016/j.heliyon.2023.e21484> (2023).
 11. Mohamed, Z. E., Using the artificial neural networks for prediction and validating solar radiation. *Journal of the Egyptian Mathematical Society*, vol. 27, no. 1, p. 47, 2019/11/28 2019, doi: 10.1186/s42787-019-0043-8 (2019).