Implementation of artificial intelligence for prediction performance of solar thermal system

Mohd Danish Irfan Mohd Sufian[1], Nur Ashida Salim[1], Hasmaini Mohamad[1], Zuhaila Mat Yasin[2]
[1]Power System Planning and Operations Research Group (PoSPO), Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia
[2]Power System Operation Computational Intelligence Research Group (POSC), Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia

Article Info

Article history:

Received Mar 30, 2022
Revised Jun 13, 2022
Accepted Jun 27, 2022

Keywords:

Artificial intelligence
Artificial neural network
Levenberg-Marquardt
Solar thermal systems

ABSTRACT

A related input parameter is used in this case study to forecast solar thermal systems (STS) capabilities and to compare which artificial neural network (ANN) algorithms and other artificial intelligence (AI) methods have the most reliable predictor for STS performance. In order to gauge the performance of the STS, this research aims to implement AI for predicting STS performance by comparing the ANN technique with other methods. Three different training algorithms which are Levenberg-Marquardt (LM), scaled conjugate gradient (SCG) and Bayesian regularization (BR) are considered in this research. This research will identify acceptable parameters and the best AI technique to use in predicting the STS performance. Previous research on STS demonstrates that the efficiency of STS has been estimated using different input parameters. The results show that the prediction of the LM training algorithm is the best for STS performance.

Corresponding Author:

Nur Ashida Salim
Faculty of Electrical Engineering, Universiti Teknologi MARA
40450 Shah Alam, Selangor, Malaysia
Email: nurashida606@uitm.edu.my

1. INTRODUCTION

Solar thermal systems (STS), which are currently satisfying the growing demand for electricity, are efficient and environment-friendly facilities [1]. Thus, methods based on intelligent machines, such as an artificial neural network (ANN) and other artificial intelligence (AI) methods, are vital in accurate and rapid machine results prediction. Previous researcher that done some analysis on STS reveals that various input parameters have been used to forecast the STS output [2]. In addition, researchers have also used a minimum number of ANN modelling training algorithms, which are widely used for training Levenberg-Marquardt (LM) algorithms [3]-[7]. Due to the research gap on STS, this paper proposes to classify the relevant input parameters to predict the performance of the STS and to compare which ANN algorithms and other AI methods give the most accurate prediction for STS performance.

2. SOLAR THERMAL SYSTEMS

The heat provided by STS may be used for practical purposes such as water or space heating. Unlike solar panels, which convert solar energy into electricity, the STS converts sunlight into heat. A solar thermal device is a form of solar collector that converts sunlight into heat. To draw sunlight, solar water heating systems usually use collectors installed on the roof of a building or home [8]. STS convert sunlight into thermal or electrical energy and are used in both commercial and residential buildings [9], [10].

3. ARTIFICIAL INTELLIGENCE TECHNIQUES

3.1. Artificial neural network

The cell body serves as a generator, the synapse functions as a reference, and the axon sends feedback signals to other neurons and carries out nonlinear operations. Figure 1 specifically shows the components of a neuron which is made up of dendrites, a cell body or soma, and the axon. Common biological neurons are individual cells, each composed of a major body of cells along with many suckers that extend from that body. The body, or soma, houses machinery for maintaining basic cell function and energy processing. There are two types of tendrils which are dendrites which receive information from other neurons and carry it to the cell body, and axons, which send information from the cell body to other neurons. ANN have been used by a few previous researches to represent and forecast STS performance. The use of the ANN technique has been very common in the field of thermal engineering systems in the last two decades [11]-[18].

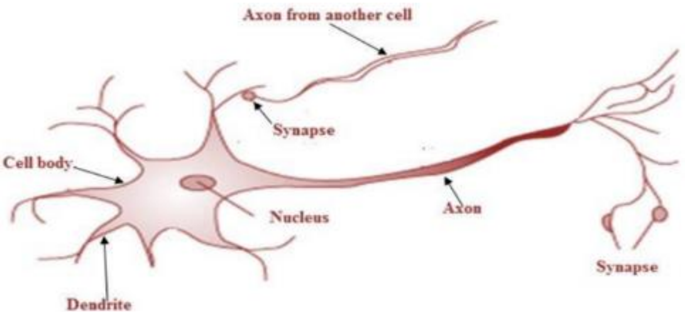


Figure 1. Basic structure of biological neurons

3.2. Type of artificial neural network training algorithm

In terms of memory requirements, processor speed, and numerical precision, all the parameters have different characteristics and performances. In a neural network, the method used to perform a learning process is called an optimisation algorithm. It is very challenging to establish which training algorithm is the quickest and optimal for a given problem [19].

3.2.1. Levenberg-Marquardt training algorithm

The teaching method of LM is derived from the Newton method [20]. The algorithm has a quicker integration of second orders, fewer iterations, and the Hessian matrix need not be computed. For certain network models that have a small number of parameters, the algorithm has a faster training speed. Let *N* be the weight and bias vector of each layer in the iterative training provided by (1). Delta *N* is the change in *N*, and changing *N* involves controlling the weights and thresholds of each layer in the network; eventually, the objective of network training is accomplished [21].

$$N = \begin{bmatrix} W_{11} & W_{12..}W_{IH} & \theta J(1) \dots \theta J(H) \\ V_{11} & V_{12..}V_{IH} & \theta I(1) \dots \theta I(0) \end{bmatrix} \tag{1}$$

$$P(N) = \sum_{i=1}^{I} e_i(N)^2 \tag{2}$$

Where, the expression function of *P*(*N*)=$e_i(N)^2$, *i=1→I* represent the square of error. *J(N)* is the Jacobian matrix given by (3), where *E(N)* is expressed as e_i *(N)*, *i=1→I* constituent vectors, and $S(N)$ is the error function given by (4). Since the LM algorithm refines the Gauss-Newton process, *ΔN* can be represented by (5).

$$J(N) = \begin{bmatrix} \frac{\partial e_1(N)}{\partial N_1} & \frac{\partial e_1(N)}{\partial N_2} & \frac{\partial e_1(N)}{\partial N_i} \\ \frac{\partial e_2(N)}{\partial N_1} & \frac{\partial e_2(N)}{\partial N_2} & \frac{\partial e_2(N)}{\partial N_i} \\ \frac{\partial e_I(N)}{\partial N_1} & \frac{\partial e_I(N)}{\partial N_1} & \frac{\partial e_I(N)}{\partial N_i} \end{bmatrix} \tag{3}$$

$$S(N) = \sum_{i=1}^{I} e_i(N) \nabla^2 e_i(N) \tag{4}$$

$$\Delta N = J^T(N)E(N)[J^T(N)J(N) + \mu I]^{-1} \tag{5}$$

Where, $\mu > 0$ is a constant and I is the unit matrix.

If *μ=0* is used for the Gauss-Newton method, LM follows the small-step gradient descent (GD) method, as it is larger. The adjustment factor is changed from μ to α. μ is increased or decreased if the training fails. Due to its positive value, the solution of *ΔN* still remains. The LM algorithm is equivalent to the system of Gauss-Newton. The measures of the LM algorithm can be described by the following steps:

- Initialise the permissible values, coefficients, thresholds, error training weights, and $k=0$ (k is the number of iterations).
- Calculate the network output *P(N)* and Jacobian matrix *J(N)* expression functions using (2) and (3), respectively.
- Using (5) to calculate *ΔN*.
- The simulations are terminated if *P(N)<ε*, otherwise, use *N+Δ* as the weights and thresholds to recalculate the function of expression *P(N)*. Let *μ=μ/α* and $p=p+1$ and return to 2) when *P(N)* is less than the *P(N)* of 2); otherwise, set μ=μ and return to 3).

3.2.2. Scaled conjugate gradient training algorithm

For significant problems, the scaled conjugate gradient (SCG) training method can be useful. It utilises second-order knowledge sans the process of line-search. The level of memory consumption can, therefore, be minimised by cutting down the volume of computed gradient information [22].

- Select the weight vector $\widetilde{w}_1$ and scalars $0 < \sigma \leq 10^{-4}, 0 < \lambda_1 \leq 10^{-6}, \bar{\lambda}_1 = 0$. Let $\tilde{p}_1 = \tilde{r}_1 = -E'(\widetilde{w}_1),\ p = 1\ and\ success = true$.
- If success=true, calculate the second order using (6), (7), and (8).

$$\sigma_{p=} \frac{\sigma}{\tilde{p}_p} \tag{6}$$

$$\tilde{S}_{p=} \frac{[E'(\widetilde{w}_1 + \sigma_p \tilde{p}_p) - E'(\widetilde{w}_1)]}{\sigma_p} \tag{7}$$

$$\delta_p = \tilde{p}_p^T \tilde{S}_p \tag{8}$$

- If the steepest direction of descent is $\tilde{r}_p \neq \tilde{0}$, set $p = p$+1 and go to 2); otherwise, terminate and return $\widetilde{w}_{p+1}$ as the minimum required.

3.2.3. Bayesian regularization training algorithm

The algorithm operates effectively and illustrates greater efficacy in preventing limited local issues through extraction of weights from the network. A robust estimation of noise and complexity is given by the Bayesian regularization (BR) algorithm. In the BR algorithm, cross-validation is unnecessary for validation purposes, which avoids part of the training results.

The E(w) gradient and (9) function are considered the first derivatives of the error function. The LM algorithm developed by Kenneth Levenberg and Donald Marquardt should be used by ANNs that deal with modest-sized problems. The second E(w) derivative is the Hessian of E(w) in (10). Wilamowski and Yu introduced the Hessian approximation, as demonstrated in (11). The LM algorithm update rule is provided as (12) [23].

$$\nabla E(w) = \frac{d}{dw} E(w) = \begin{bmatrix} \frac{\partial E}{\partial w1} \\ \frac{\partial E}{\partial w2} \\ \cdots \\ \frac{\partial E}{\partial wN} \end{bmatrix} \tag{9}$$

$$\mathrm{H} = \nabla\nabla E(w) = \frac{d^2}{dw^2} E(w) = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w1 \partial w2} \cdots\cdots & \frac{\partial^2 E}{\partial w1 \partial wN} \\ \frac{\partial^2 E}{\partial w2 \partial w1} & \frac{\partial^2 E}{\partial w_2^2} \cdots\cdots & \frac{\partial^2 E}{\partial w2 \partial wN} \\ \frac{\partial^2 E}{\partial wN \partial w1} & \frac{\partial^2 E}{\partial wN \partial w2} \cdots\cdots & \frac{\partial^2 E}{\partial w_N^2} \end{bmatrix} \tag{10}$$

$$\mathrm{H} = J^T J + \mu I \tag{11}$$

$$W_{k+1} = w_k - (J^T J + \mu I)^{-1} J_k e_K \tag{12}$$

Where *J*: Jacobian matrix (matrix of first derivatives with respect to weight vector), *μ*: combination coefficient, and *I*: identity matrix.

3.3. Linear regression

Regression analysis is a mathematical method in which the association between variables is investigated and modelled. The basic linear regression model is a model that has one independent variable, x, which is related to one answer variable, y, which is a straight line. This simple linear regression model is given by (13).

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{13}$$

Where an undefined constant is the intercept, β_0, and the slope, β_1, and a random error is ε. It is assumed that the errors have a mean of zero and an undefined variance, σ^2. The β_0 and β_1 parameters are undefined and must be calculated using sample data. The basic theorem of linear regression is sometimes called the Least Squares Regression equation. This shows the criteria used to select the best fitting rows, in which the number of residual squares should be the least. As such, as shown in (14), the least squares regression is the line for which the sum of squared residuals is minimum [24].

$$\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \tag{14}$$

3.4. Support vector machine

In 1992, Vapnik first implemented the support vector machine (SVM) as a learning concept for classifying pattern recognition. Capacity analysis places SVM as the latest method in pattern recognition in numerous implementations and it is a theme that is growing rapidly in popularity. SVM is a learning machine method that functions based on the structural risk minimisation (SRM) principle [25]-[29]. SVM also provides improved classification outcomes and are commonly used for methods of pattern detection, such as optimum probability and classifiers of the neural network. Radial basis function and multilayer perceptron classifiers resolve quadratic programming (QP).

The learning problem in SVM setting is as follows: between some high-dimensional input vector, x, and scalar output, y; there is some undefined and nonlinear dependence (mapping, function) y=f(x) (or the vector output, y, as in the case of multiclass SVM [30]. The structure of the optimised SVM is shown in Figure 2. In SVM, using nonlinear mapping, the historical data of the time series is transformed into higher-dimensional space [31]-[33].

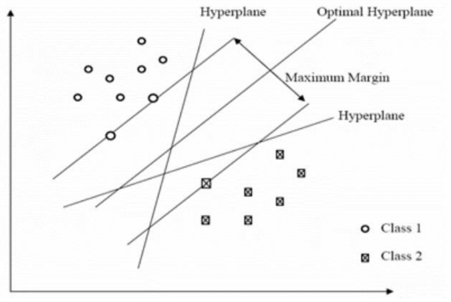


Figure 2. Structure of the optimised SVM

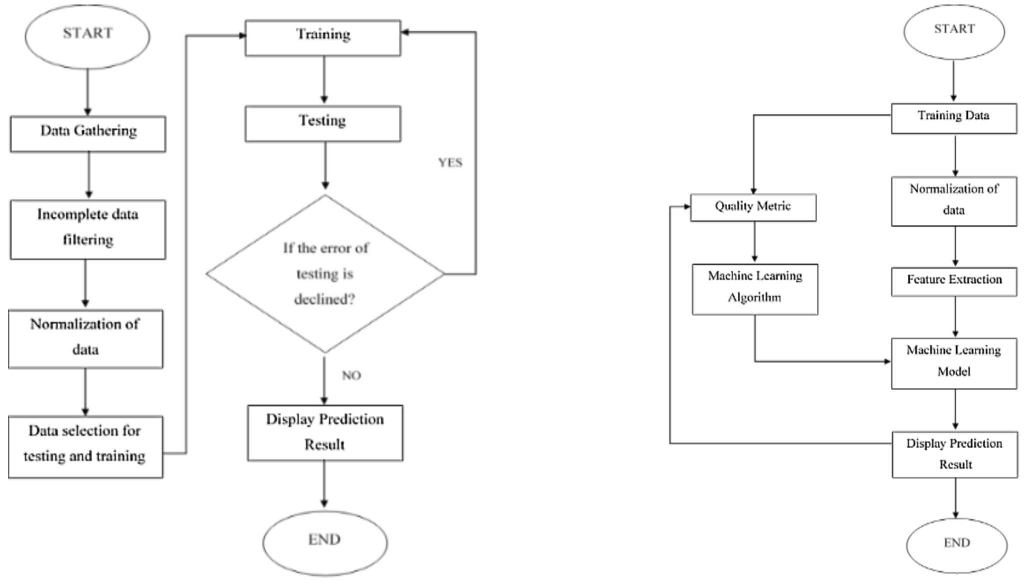
4. METHOD

In this paper, ANN will be designed to obtain the efficiency of the STS. MATLAB software with the aid of certain techniques and applications will be utilised. Additionally, to validate the proposed technique, other AI methods will also be developed.

4.1. Flowchart and block diagram of the project

The ultimate process of ANN prediction is shown in Figure 3(a). Other than that, training performance for various ANN training algorithms and training percentages is also part of the process. In the beginning of the process, data are gathered and undergo normalisation. Then, the selected data undergo training and testing process. The simulation will display the prediction results upon no testing error.

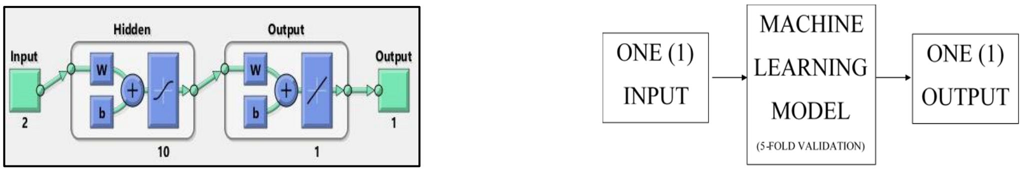
Figure 3(b) shows the machine learning flowchart for the LR and SVM methods used to predict the performance of STS. Initially, the training data are normalised before some features are extracted, resulting in the machine learning model, which are LR and SVM. Next, the prediction result is compared with actual data to establish the error by quality matric.


START
Data Gathering
Incomplete data filtering
Normalization of data
Data selection for testing and training
Training
Testing
If the error of testing is declined?
YES
NO
Display Prediction Result
END
START
Training Data
Normalization of data
Feature Extraction
Machine Learning Model
Display Prediction Result
END
Quality Metric
Machine Learning Algorithm

(a) (b)

Figure 3. Block diagram of (a) artificial neural network prediction process and (b) machine learning

Figure 4(a) shows that the ANN block diagram consists of two input layers with 10 hidden layers and one output layer for LM, SCG, and BR training algorithm. Figure 4(b) shows the linear regression and SVM block diagram, which consists of one input and output with five-fold validation. This research used one input parameter, which is temperature, while the output that needs to be predicted is relative humidity (%). The fold validation is set to five to provide optimum prediction output. If fold validation is set at more than five, the difference in size between the training set becomes smaller and will affect the prediction performance.



(a) (b)

Figure 4. Block diagram of (a) ANN method and (b) linear regression and SVM methods

4.2. Normalised data

Every prediction conducted in this research had utilised normalised data in order to give an optimal result with the range of normalisation 0 to 1. The data are taken from an article by Anastasia Mylona, which is modelling the thermal performance of complex glazing systems [34]. In (15) is used to normalise the data.

$$X_i = 0.9 \times \frac{X_{ni} - min(Xn)}{max(Xn) - min(Xn)} + 0.1 \tag{15}$$

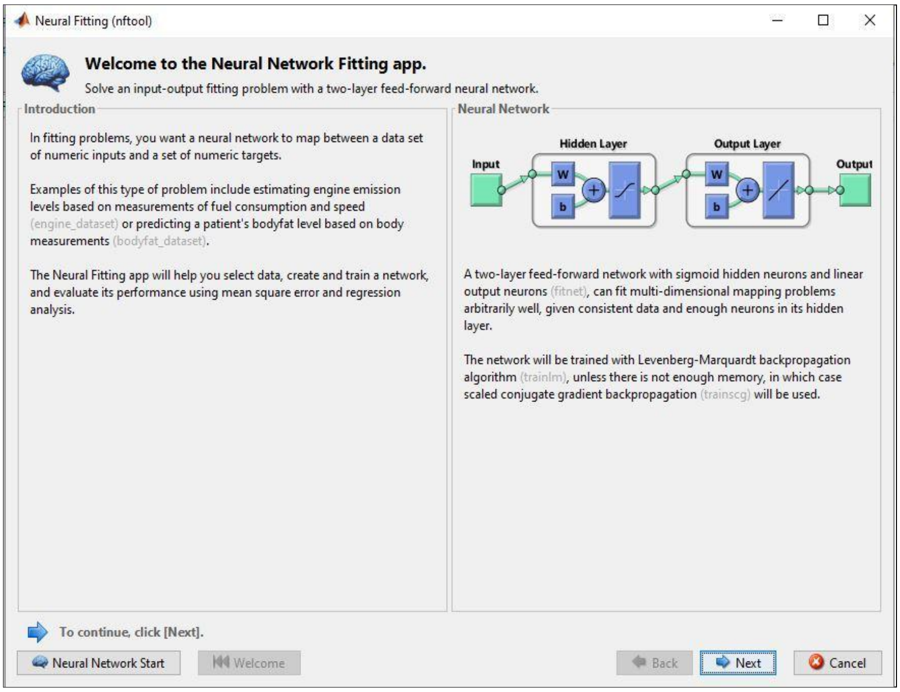
Where *Xi*=normalised data, *Xni*=data that need to be normalised, *Min (Xn)*=minimum value in data set, and *Max (Xn)*=maximum value in data set.

4.3. Software development

The MATLAB software was used in the whole process of prediction, in which the MATLAB 2019b version was used to create programming based on the project. The MATLAB software has two media that were used in this project, which are programming script and application tools such as neural net fitting tool (nftool) and regression learner application.

4.3.1. Neural net fitting tool

The nftool shown in Figure 5 is used to address the issue of data fitting, used together with LM, BR, and SCG to train the two-layer feed-forward network [35]. By using the nftool, the neural fitting will assist in selecting data, create and train a network. It also assists in the evaluation of its performance using mean square error and regression technique.



Neural Fitting (nftool)

Welcome to the Neural Network Fitting app.

Solve an input-output fitting problem with a two-layer feed-forward neural network.

Introduction

In fitting problems, you want a neural network to map between a data set of numeric inputs and a set of numeric targets.

Examples of this type of problem include estimating engine emission levels based on measurements of fuel consumption and speed (engine_dataset) or predicting a patient's bodyfat level based on body measurements (bodyfat_dataset).

The Neural Fitting app will help you select data, create and train a network, and evaluate its performance using mean square error and regression analysis.

Neural Network

A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons (fitnet), can fit multi-dimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer.

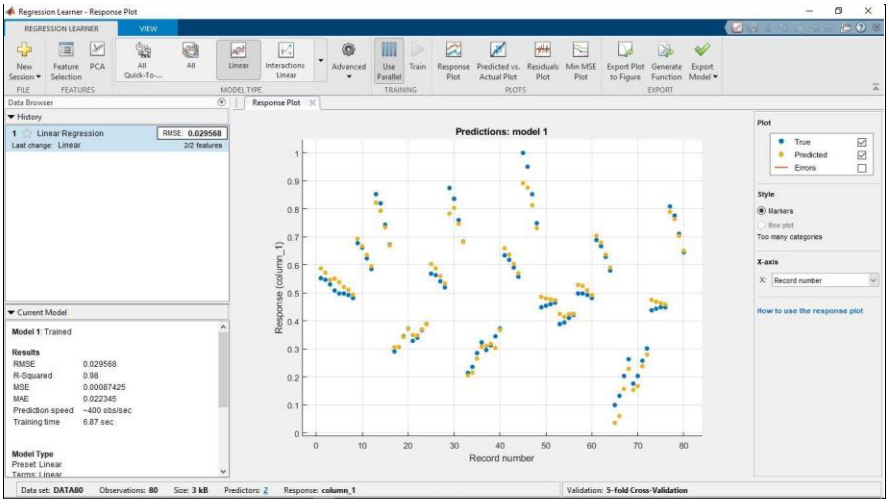
The network will be trained with Levenberg-Marquardt backpropagation algorithm (trainlm), unless there is not enough memory, in which case scaled conjugate gradient backpropagation (trainscg) will be used.

To continue, click [Next].

Figure 5. Neural nftool

4.3.2. Regression learner application

Figure 6 shows that MATLAB used the regression learner application to help select between different algorithms to train and test regression models, compare their validation errors side-by-side after practicing several models, and select the better prediction performance in terms of mean squared error (MSE), root-mean-square error (RMSE), and training time. It also assists in deciding which algorithm to use. In regression learner application, Figure 7 displays the flow map for training regression models [36].



History

1 Linear Regression RMSE: 0.029568
Last change: Linear 2/2 features

Current Model

Model 1: Trained

Results

RMSE	0.029568
R-Squared	0.98
MSE	0.00087425
MAE	0.022345
Prediction speed	~400 obs/sec
Training time	6.87 sec

Model Type
Preset: Linear
Terms: Linear

Data set: DATA80 Observations: 80 Size: 3 kB Predictors: 2 Response: column_1 Validation: 5-fold Cross-Validation

Figure 6. Regression learner application

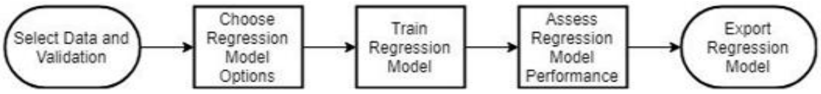

Select Data and Validation
Choose Regression Model Options
Train Regression Model
Assess Regression Model Performance
Export Regression Model

Figure 7. Flowchart of training regression models

5. RESULTS AND DISCUSSION

The prediction result is divided into three sections: first, the comparative prediction performance of STS with the ANN training algorithms, such as LM, SCG, and BR. The second section compares the prediction performance of STS with ANN training algorithm and different training percentages, in which the training percentage was increased from 50% to 70% for all ANN training algorithms: LM, SCG, and BR. Finally, a comparison was made on the prediction performance of STS with artificial neural network (ANN) and other AI methods, such as LR and SVM.

5.1. Simulation outputs

5.1.1. Comparative study of artificial neural network with different training algorithm

Based on Table 1, the LM training algorithm shows the best regression value of almost 1, which is at 0.99. At the same time, the algorithm training time is faster compared with BR and SCG. BR training algorithm shows that the regression value is almost like that of LM.

Table 1. Comparison of errors, regression coefficient, and training time by different ANN algorithms

Training algorithm	RMSE (% of RH)	MSE (% of RH)	Regression coefficient (R^2)	Time elapsed (s)
LM	0.0011	0.000064702	0.999	0.41648
BR	0.00080072	NaN	0.99939	1.395452
SCG	0.0049	0.0008278	0.97613	0.420273

Note: MSE for BR training algorithm cannot be calculated using the software

5.1.2. Comparative study of artificial neural network training algorithm with different training percentage

According to Table 2, the LM training algorithm with 70% training percentage is the best prediction with regression value 0.999. The RMSE for 70% training percentage is considered the least value of RMSE, which is 0.0011. Meanwhile, 50% training percentage has the regression value of 0.9985 with an RMSE value of 0.0013. The training time for 70% training percentage is longer than that of 50% training percentage, where the LM with 70% training percentage gave the best prediction results. Finally, the 70% of SCG training algorithm is deemed the best prediction due to its regression value of 0.97613, which is close to 1, as compared to that given by the 50% training percentage. Overall, in this section, the LM training algorithm with 70% training percentage provided the best prediction compared with others.

Table 2. Comparison of errors, regression coefficient, and training time by different training percentages

Training algorithm	Training percentage (%)	RMSE (% of RH)	MSE (% of RH)	Regression coefficient (R^2)	Time elapsed (s)
LM	50	0.0013	0.00029229	0.99856	0.390622
	70	0.0011	0.000064702	0.999	0.41648
BR	50	0.00079409	NaN	0.99941	0.861464
	70	0.00080072	NaN	0.99939	1.395452
SCG	50	0.0063	0.0052	0.96349	0.420051
	70	0.0049	0.0008278	0.97613	0.420273

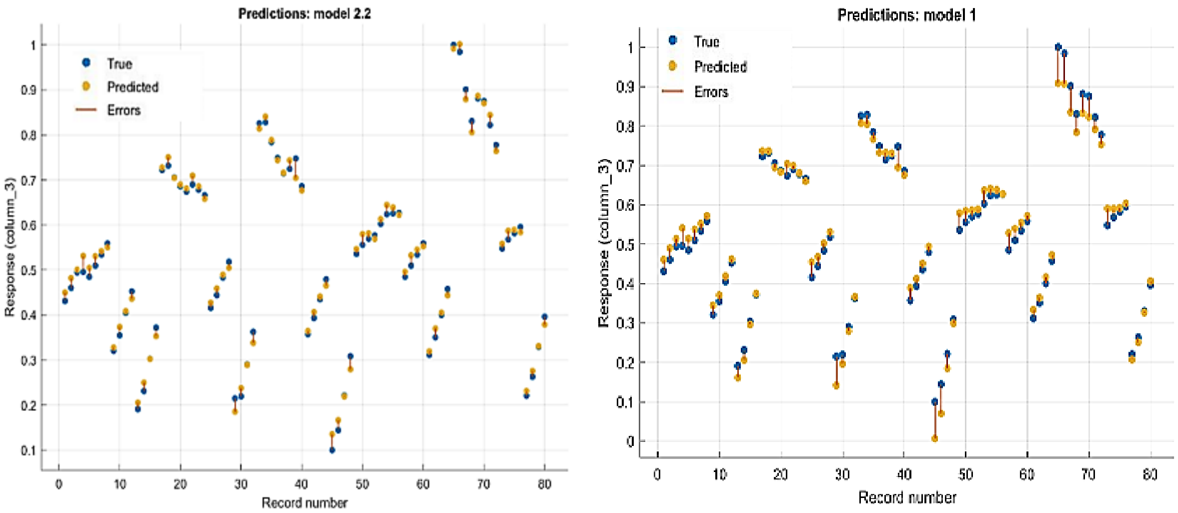
5.1.3. Comparative study of artificial neural network with other artificial intelligence methods

Table 3 the best prediction in this section is ANN, with a regression value of 0.999. The values of both errors, RMSE and MSE, are lower than that of other methods, which are linear regression and SVM; in which their training time is faster compared with others. If comparison is made between linear regression and SVM, it will demonstrate that SVM provides the best prediction even with longer training time compared with linear regression.

As shown in Figures 8(a) and 8(b), the linear regression plot errors are more significant, causing the lower accuracy of prediction, while in SVM, the error is least and thus, the accuracy of prediction is higher. Figures 9(a) and 9(b) show the comparison plot between the actual value and predicted value for linear regression and SVM, respectively; the blue colour is the expected value.

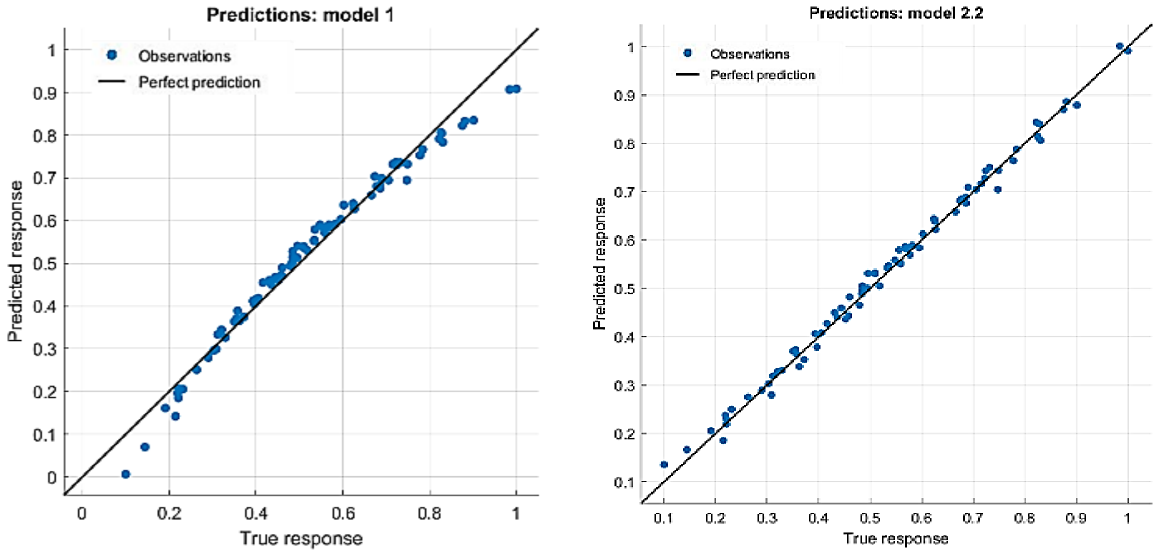
Table 3. Comparison of errors, regression coefficient, and training time by different artificialintelligence methods

Training algorithm	RMSE (% of RH)	MSE (% of RH)	Regression coefficient (R^2)	Time elapsed (s)
Linear regression	0.0031733	0.001007	0.98	0.79793
SVM	0.0015825	0.00025044	0.99	4.1361
ANN	0.0011	0.000064702	0.999	0.41648



(a) (b)

Figure 8. Response plot of (a) linear regression and (b) SVM



(a) (b)

Figure 9. Predicted vs actual plot of (a) linear regression and (b) SVM

6. CONCLUSION AND RECOMMENDATION

This work concerns the application of AI for STS prediction performance. This analysis demonstrated one of the better training algorithms, which typically had a value similar to the actual value from the predicted training algorithm of LM. In the second section, the contrast with a training percentage between the training algorithm of ANN shows that the percentage of preparation has influenced the prediction results. The better approaches of AI from this section are the ANN relative to other techniques, which are linear regression and SVM. The ANN produces superior results in the prediction of STS performance. Only then can further analyses of different hidden layers of every training algorithm can be developed in the future.

ACKNOWLEDGEMENTS

The authors would like to thank the Research Management Institute (RMI), Universiti Teknologi MARA, Malaysia, through research grant 600-RMC/YTR/5/3 (011/2020) for the financial support of this research.

REFERENCES

[1] A. Sözen, T. Menlik, and S. Ünvar, "Determination of efficiency of flat-plate solar collectors using neural network approach," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1533–1539, 2008, doi: 10.1016/j.eswa.2007.08.080.

[2] H. K. Ghritlahre, P. Chandrakar, and A. Ahmad, "Application of ANN model to predict the performance of solar air heater using relevant input parameters," *Sustainable Energy Technologies and Assessments*, vol. 40, 2020, doi: 10.1016/j.seta.2020.100764.

[3] A. F. Mashaly and A. A. Alazba, "MLP and MLR models for instantaneous thermal efficiency prediction of solar still under hyper-arid environment," *Computers and Electronics in Agriculture*, vol. 122, pp. 146–155, 2016, doi: 10.1016/j.compag.2016.01.030.

[4] M. Mohanraj, S. Jayaraj, and C. Muraleedharan, "Applications of artificial neural networks for thermal analysis of heat exchangers-a review," *International Journal of Thermal Sciences*, vol. 90, pp. 150–172, 2015, doi: 10.1016/j.ijthermalsci.2014.11.030.

[5] M. Caner, E. Gedik, and A. Keĉebaŝ, "Investigation on thermal performance calculation of two type solar air collectors using artificial neural network," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1668–1674, 2011, doi: 10.1016/j.eswa.2010.07.090.

[6] H. K. Ghritlahre and R. K. Prasad, "Exergetic performance prediction of a roughened solar air heater using artificial neural network," *Journal of Mechanical Engineering*, vol. 64, no. 3, pp. 195–206, 2018, doi: 10.5545/sv-jme.2017.4575.

[7] E. Arcaklioglu, A. Erişen, and R. Yilmaz, "Artificial neural network analysis of heat pumps using refrigerant mixtures," *Energy Conversion and Management*, vol. 45, no. 11–12, pp. 1917–1929, 2004, doi: 10.1016/j.enconman.2003.09.028.

[8] A. Kylili, P. A. Fokaides, A. Ioannides, and S. Kalogirou, "Environmental assessment of solar thermal systems for the industrial sector," *Journal of Cleaner Production*, vol. 176, pp. 99–109, 2018, doi: 10.1016/j.jclepro.2017.12.150.

[9] R. Sioshansi and P. Denholm, "The value of concentrating solar power and thermal energy storage," *IEEE Transactions Sustainable Energy*, vol. 1, no. 3, pp. 173–183, Oct. 2010, doi: 10.1109/TSTE.2010.2052078.

[10] S. Haykin, *Neural networks: a comprehensive foundation*. Hoboken, NJ, USA: Prentice Hall, 2018.

[11] H. K. Ghritlahre and R. K. Prasad, "Application of ANN technique to predict the performance of solar collector systems–a review," *Renewable and Sustainable Energy Reviews*, vol. 84, pp. 75–88, 2018, doi: 10.1016/j.rser.2018.01.001.

[12] H. K. Ghritlahre and R. K. Prasad, "Prediction of heat transfer of two different types of roughened solar air heater using Artificial Neural Network technique," *Thermal Science and Engineering Progress*, vol. 8, pp. 145-153, 2018, doi: 10.1016/j.tsep.2018.08.014.

[13] O. Kisi, M. Tombul, and M. Z. Kermani, "Modeling soil temperatures at different depths by using three different neural computing techniques," *Theoretical and Applied Climatology*, vol. 121, pp. 377–387, 2015, doi: 10.1007/s00704-014-1232-x.

[14] H. K. Ghritlahre and R. K. Prasad, "Investigation of thermal performance of unidirectional flow porous bed solar air heater using MLP, GRNN, and RBF models of ANN technique," *Thermal Science and Engineering Progress*, vol. 6, pp. 226-235, 2018, doi: 10.1016/j.tsep.2018.04.006.

[15] H. K. Ghritlahre and R. K. Prasad, "Exergetic performance prediction of solar air heater using MLP, GRNN and RBF models of artificial neural network technique," *Journal of Environmental Management*, vol. 223, pp. 566-575, 2018, doi: 10.1016/j.jenvman.2018.06.033.

[16] L. Wang, O. Kisi, M. Zounemat-Kermani, G. A. Salazar, Z. Zhue, and W. Gong, "Solar radiation prediction using different techniques: model evaluation and comparison," *Renewable and Sustainable Energy Reviews*, vol. 61, pp. 384-397, 2016, doi: 10.1016/j.rser.2016.04.024.

[17] H. K. Ghritlahre, R. K. Prasad, and A. Ahmad, "A comprehensive review on performance prediction of solar air heater using ANN," *Annals of Data Science*, vol. 8, pp. 405-449, 2019, doi: 10.1007/s40745-019-00236-1.

[18] A. Ahmad, H. K. Ghritlahre, and P. Chandrakar, "Implementation of ANN technique for performance prediction of solar thermal systems: a comprehensive review," *Trends in Renewable Energy*, vol. 6, no. 1, pp. 12-36, 2020, doi: 10.17737/tre.2020.6.1.00110.

[19] M. R. Mustafa, R. B. Rezaur, S. Saiedi, H. Rahardjo, and M. H. Isa, "Evaluation of MLP-ANN training algorithms for modeling soil pore-water pressure responses to rainfall," *Journal of Hydrologic Engineering*, vol. 18, no. 1, pp. 50–57, 2013, doi: 10.1061/(ASCE)HE.1943-5584.0000599.

[20] Z. Min, L. Xiao, L. Cao, and H. Yan, "Application of the neural network in diagnosis of breast cancer based on levenberg-marquardt algorithm," *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, 2017, pp. 268-272, doi: 10.1109/SPAC.2017.8304288.

[21] A. D. Kulkarni and C. D. Cavanaugh, "Fuzzy neural network models for classification," *Applied Intelligence*, vol. 12, pp. 207-215, 2000, doi: 10.1023/A:1008367007808.

[22] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, pp. 525-533, 1993, doi: 10.1080/10106049908542127.

[23] K. Jazayeri, M. Jazayeri, and S. Uysal, "Comparative analysis of levenberg-marquardt and bayesian regularization backpropagation algorithms in photovoltaic power estimation using artificial neural network," *ICDM 2016: Advances in Data Mining. Applications and Theoretical Aspects*, 2016, pp. 80-95, doi: 10.1007/978-3-319-41561-1_7.

[24] S. Ibrahim, I. Daut, Y. M. Irwan, M. Irwanto, N. Gomesh, and Z. Farhana, "Linear regression model in estimating solar radiation in Perlis," *Energy Procedia*, vol. 18, 2012, doi: 10.1016/j.egypro.2012.05.156.

[25] X. Zhang, "Introduction to statistical learning theory and support vector machine (in Chinese)," *Acta Automatica Sinica*, vol. 26, no. 1, Jan. 2000.

[26] V. N. Vapnik, *Statistical learning theory*. Hoboken, NJ, USA: John Wiley & Sons, 1998.

[27] V. N. Vapnik, "An overview of statistical learning theory," in *IEEE Trans. Neural Network*, vol. 10, no. 5, pp. 988-999, 1999, doi: https: 10.1109/72.788640.

[28] B. Liu, Z. -F. Hao, J. Lu and S. -Q. Liu, "Apply support vector machine for CRM problem," *2007 International Conference on Machine Learning and Cybernetics*, 2007, pp. 3288-3292, doi: 10.1109/ICMLC.2007.4370715.

[29] S. Lu, J. Meng and G. Cao, "Support vector machine based on a new reduced samples method," *2010 International Conference on Machine Learning and Cybernetics*, 2010, pp. 1510-1514, doi: 10.1109/ICMLC.2010.5580828.

[30] Nuraedah, M. Bakri, and A. A. Kasim, "Quadratic support vector machine for the bomba traditional textile motif classification,"

Indonesian Journal of Electrical Engineering and Computer Science, vol. 11, no. 3, pp. 1004-1014, 2018, doi: 10.11591/ijeecs.v11.i3.pp1004-1014.

[31] J. Umamaheswari and G. Radhamani, "Quadratic program optimization using support vector machine for CT brain image classification," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 4, pp. 305-309, 2012.

[32] J. Zeng and W. Qiao, "Short-term solar power prediction using a support vector machine," *Renewable Energy*, vol. 52, pp. 118-127, 2013, doi: 10.1016/j.renene.2012.10.009.

[33] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, pp. 293-300, 1999, doi: 10.1023/A:1018628609742.

[34] A. Mylona, "Modelling the thermal performance of complex glazing systems," Ph.D. thesis, Welsh School of Architecture, Cardiff University, Cardiff, United Kindom, 2007.

[35] A. K. Yadav, H. Malik, and A. P. Mittal, "Artificial neural network fitting tool based prediction of solar radiation for identifying solar power potential," *Journal of Electrical Engineering*, vol. 15, no. 2, pp. 25–29, 2015.

[36] P. S. Popolo and A. M. Johnson, "Relating cepstral peak prominence to cyclical parameters of vocal fold vibration from high-speed videoendoscopy using machine learning: a pilot study," *Journal of Voice*, vol. 35, no. 5, pp. 703-716, 2021, doi: 10.1016/j.jvoice.2020.01.026.

BIOGRAPHIES OF AUTHORS

Mohd Danish Irfan Mohd Sufian iD 8 SC P received Bachelor in Electrical Engineering (Hons.) from Universiti Teknologi MARA in 2021 and Diploma in Electrical Engineering from Universiti Teknologi MARA in 2017. His research includes renewable energy, power system, and related areas. He can be contacted at email: danishsufian97@gmail.com.

Nur Ashida Salim iD 8 SC P received her Ph.D. in Electrical Engineering from Universiti Teknologi MARA in 2015, Master's in engineering (power system and electrical energy) from Universiti Malaya in 2006 and Bachelor's in electrical engineering (Hons.) from Universiti Teknologi MARA in 2003. She is currently an Associate Professor at the Centre for Electrical Power Engineering Studies, Faculty of Electrical Engineering, Universiti Teknologi MARA. Her research includes power system reliability, power system planning, power system stability, power system asset management and other related areas. To date, she has published more than 40 journal articles and many proceeding papers. She can be contacted at email: nurashida606@uitm.edu.my.

Hasmaini Mohamad iD 8 SC P received the B.Eng., M.Eng., and Ph.D. degrees from the University of Malaya, in 1999, 2004, and 2013, respectively. She started her career as a lecturer in Universiti Teknologi MARA in 2003 where currently she is an Associate Professor at the Centre for Electrical Power Engineering Studies, Faculty of Electrical Engineering. Apart from that, she has published more than 50 journal papers including high impact ISI journals and 20 conference papers. Her major research interest includes islanding operation of distributed generation, hydro generation, load sharing technique, and load shedding scheme. She can be contacted at email: hasmaini@uitm.edu.my.

Zuhaila Mat Yasin iD 8 SC P graduated from Universiti Sains Malaysia with honours degree in Electrical and Electronics Engineering in 1998. She obtained her MSc degree in 2008 and Ph.D. degree in 2015 from Universiti Teknologi MARA. She is currently a senior lecturer at Universiti Teknologi MARA. Her research interest includes power system operation, optimisation, distributed generation, artificial intelligence, and smart grid system. She can be contacted at email: zuhai730@uitm.edu.my.