Comparative investigation of artificial neural network learning algorithms for modeling solar still production

Ahmed F. Mashaly and A. A. Alazba

ABSTRACT

Three artificial neural network learning algorithms were utilized to forecast the productivity (MD) of a solar still operating in a hyper-arid environment. The learning algorithms were the Levenberg–Marquardt (LM), the conjugate gradient backpropagation with Fletcher–Reeves restarts, and the resilient backpropagation. The Julian day, ambient air temperature, relative humidity, wind speed, solar radiation, temperature of feed water, temperature of brine water, total dissolved solids (TDS) of feed water, and TDS of brine water were used in the input layer of the developed neural network model. The MD was located in the output layer. The developed model for each algorithm was trained, tested, and validated with experimental data obtained from field experimental work. Findings revealed the developed model could be utilized to predict the MD with excellent accuracy. The LM algorithm (with a minimum root mean squared error and a maximum overall index of model performance) was found to be the best in the training, testing, and validation stages. Relative errors in the predicted MD values of the developed model using the LM algorithm were mostly in the vicinity of $\pm 10\%$. These results indicated that the LM algorithm is the most ideal and accurate algorithm for the prediction of the MD with the developed model.

Key words | artificial neural network, learning algorithm, Levenberg–Marquardt, prediction, solar still

Ahmed F. Mashaly (corresponding author)
A. A. Alazba
Alamoudi Chair for Water Researches,
King Saud University,
P.O. Box, 2460,
Riyadh 11451,
Saudi Arabia
E-mail: *mashaly.ahmed@gmail.com*

A. A. Alazba
Agricultural Engineering Department,
King Saud University,
P.O. Box, 2460,
Riyadh 11451,
Saudi Arabia

NOMENCLATURE

ANN	artificial neural network	RMSE	root mean square error
BP	backpropagation	RP	resilient backpropagation
CGF	conjugate gradient backpropagation with Fletcher–Reeves restarts	Rs	solar radiation
		x_{max}	maximum observed value
CRM	coefficient of residual mass	x_{min}	minimum observed value
E	efficiency coefficient	$x_{o,i}$	observed value
EC	electrical conductivity	$x_{p,i}$	predicted value
g_k	current gradient	$\bar{x}_o$	averaged observed values
JD	Julian day	p	momentum parameter
LM	Levenberg–Marquardt	TDS_F	total dissolved solids of feed water
MD	productivity capacity	TDS_B	total dissolved solids of brine water
MLP	multilayer perceptron	To	ambient temperature
OI	overall index of model performance	T_F	temperature of feed water
pH	power of hydrogen	T_B	temperature of brine water
R	correlation coefficient	U	wind speed
R^2	coefficient of determination	$w_{i,j}$	weights between input and hidden layers
RH	relative humidity	w_j	weights between hidden and output layers

GREEK SYMBOLS

ρ density
Δw_k vector of weights changes
α_k learning rate

INTRODUCTION

Today, two challenges face humankind, namely water and energy. Desalination based on solar energy can present a solution to these challenges. One of the simplest devices that may be employed in solar desalination/distillation is a solar still. Solar stills utilize the sun, which is a sustainable and pollution-free source, to produce fresh water (Ayoub & Malaeb 2012). Forecasting and modeling processes of solar still production are of great importance. They contribute to the knowledge of the potential productivities attainable by the still without the effort and time to conduct various experiments. Moreover, they play a major role in decision-making processes, and through them we will predict and know whether productivity is sufficient to meet the water requirement for different purposes such as drinking or irrigation.

Classical modeling techniques are complex, need a long time for calculating and sometimes are totally unreliable (Tripathy & Kumar 2008). The use of artificial neural networks (ANNs) in the solar desalination field could produce results that are not simply obtained by classical modeling techniques (Santos *et al.* 2012). With the ANN technique, apart from decreasing the whole time required, it is possible to find solutions that make solar applications such as solar desalination more feasible and thus more attractive to potential users. They are capable of learning the key information patterns within the multi-dimensional information domain (Kalogirou 2001). The ANN technique falls within the computational intelligence generic non-linear analog techniques (Kalogirou *et al.* 2014). In the past decade, many works and studies have been published using ANNs in desalination/distillation and also in solar desalination/distillation. By way of example, not exhaustive enumeration, Gao *et al.* (2007) used ANN to analyze the seawater desalination process and presented a new approach to simulate the water production ratio of the desalination

system. Lee *et al.* (2009) developed an ANN to analyze the performance of a seawater reverse osmosis desalination plant, and it was thereafter applied to the simulation of feed water temperature. Khayet & Cojocaru (2013) utilized ANN methodology in modeling a membrane distillation performance index. Santos *et al.* (2012) investigated and evaluated the effectiveness of modeling solar still productivity using ANNs and meteorological data. Hamdan *et al.* (2013) used and compared three ANN models to find the performance of triple solar stills. Moreover, Hamdan *et al.* (2014) evaluated the sensitivity of the ANN predictions, and determined the minimum inputs required to precisely model solar still performance. Porrazzo *et al.* (2013) used an ANN model for performance analysis of a solar-powered membrane distillation system under several operating conditions. It is clear that there is much research and many studies on the use of neural networks for solar desalination and stills. On the other hand, one of the most important difficulties in modeling by neural networks is selecting the appropriate learning algorithm to obtain the best performance. Accordingly, the objectives of this study are to investigate and assess the ability of ANN to forecast solar still productivity using weather and operating parameters. In addition, comparison of different backpropagation (BP) learning algorithms will also be conducted to find out the best learning algorithm for predicting solar still productivity.

ANN

An ANN is a biologically inspired computational model composed of different processing elements, referred to as single units or artificial neurons. They are associated with coefficients or weights that establish the neural structure (Haykin 2009). When processing information, the processing elements have weighted inputs, transfer function, and outputs. There are several kinds of neural networks with various structures, however they have all been described using the transfer functions utilized in the processing elements (neurons), using the training or learning algorithm and using the connection formula. An ANN is composed of single-layer or multiple-layer neurons. Multilayer perceptron (MLP) is the best model for complicated and difficult problems, due to it overcoming the drawback of the single-layer perceptron via the addition of more

hidden layers. In a feed-forward MLP network, the input signals are multiplied by the connection weights and summed before a direct transfer function to give the output for that neuron. The transfer (activation) function is performed on the weighted sum of the neuron's inputs. Normally, the most utilized transfer function is the sigmoid function. ANN is trained with input and target pair patterns with the capability of learning, and there are several different learning algorithms. The BP learning algorithm is used for MLP (Rojas 1996). Inputs are fed forward through the ANN to optimize the weights between neurons. By backward propagation of the error through training, adjustment of the weights is made. The ANN takes the input and target values in the training data set and modifies the value of the weighted links to decrease the difference between the values of the output and target. The error is reduced through numerous training cycles referred to as epochs until the ANN reaches a specified level of accuracy. However, the number of layers, in addition to the processing elements per layer, significantly impacts the capabilities of the MLP (Alsmadi et al. 2009). A standard MLP neural network has at least three layers. The first layer is the input layer, corresponding to the problem input parameters with one node/neuron for each input parameter. The second layer is the hidden layer, utilized to capture non-linear relationships among the parameters. The third layer is the output layer, utilized to provide predicted values. In this study, the output layer has only one neuron corresponding to the forecasted result. The relationship between the output y_t and the input x_t is expressed by

$$y_t = w_o + \sum_{j=1}^{q} w_j \cdot f\left(w_{o,j} + \sum_{i=1}^{p} w_{i,j} \cdot x_t\right) \tag{1}$$

where $w_{i,j}$ ($i = 0,1,2,...,p$; $j = 1,2,...,q$) and w_j ($j = 0,1,2,...,q$) are the connection weights, p is the number of input neurons, q is the number of hidden neurons, and f is a non-linear transfer function that helps the system to learn non-linear features. In this study, the linear and tan-sigmoid transfer functions were used.

Linear transfer function (purelin)

$$f(x) = x \tag{2}$$

Tan-sigmoid transfer function (tansig)

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{3}$$

The MLP is trained utilizing the BP algorithm and the weights are optimized. The objective function to reduce the sum of the squares of the difference between the desirable output ($y_{t,p}$) and the predicted output ($y_{t,d}$) is expressed by

$$E = 0.5 \sum (y_{t,p} - y_{t,d})^2 = 0.5 \sum_{t=1} e^2 \tag{4}$$

The training of the network is achieved by the BP (Haykin 2009) algorithm trained with the steepest descent algorithm given by this equation

$$\Delta w_k = -\alpha_k \cdot g_k \tag{5}$$

where Δw_k is a vector of weights changes, g_k is the current gradient, α_k is the learning rate that determines the length of the weight update. So as to avoid oscillations and to decrease the sensitivity of the network to fast changes of the error surface (Jang et al. 1997), the change in weight is made dependent on the past weight change via adding a momentum term

$$\Delta w_k = -\alpha_k \cdot g_k + p \cdot \Delta w_{k-1} \tag{6}$$

where p is the momentum parameter. In addition, the momentum allows escaping from small local minima on the error surface (Ramírez et al. 2003).

MATERIALS AND METHODS

Experiment setup

The experiments were conducted at the Agricultural Research and Experiment Station, Department of Agricultural Engineering, King Saud University, Riyadh, Saudi Arabia (24°44′10.90″N, 46°37′13.77″E) between February and April 2013. The weather data were obtained from a weather station (model: Vantage Pro2; manufacturer: Davis, USA) close to the experimental site (24°44′12.15″N,

46°37′14.97″E). The solar still system used in the experiments was constructed from a 6 m^2 single stage of C6000 panel (F Cubed Ltd, Carocell Solar Panel, Australia). The solar still was manufactured using modern, cost-effective materials such as coated polycarbonate plastic. When heated, the panel distilled a film of water that flowed over the absorber mat of the panel. The panel was fixed at an angle of 29° from the horizontal plane. The basic construction materials were galvanized steel legs, an aluminum frame, and polycarbonate covers. The transparent polycarbonate was coated on the inside with a special material to prevent fogging (patented by F Cubed, Australia). Front and cross-sectional views of the solar still are presented in Figure 1.

The water was fed to the panel using a centrifugal pump (model: PKm 60, 0.5 HP, Pedrollo, Italy) with a constant flow rate of 10.74 L/h. The feed was supplied by eight drippers/nozzles, creating a film of water that flowed over the absorbent mat. Underneath the absorbent mat was an aluminum screen that helped to distribute the water across the mat. Beneath the aluminum screen was an aluminum plate. Aluminum was chosen for its hydrophilic properties, to assist in the even distribution of the sprayed water. Water flows through and over the absorbent mat, and solar energy is absorbed and partially collected inside the panel; as a result, the water is heated and hot air circulated naturally within the panel. First, the hot air flows toward the top of the panel, then reverses its direction to approach the bottom of the panel. During this process of circulation, the humid air touches the cooled surfaces of the transparent polycarbonate cover and the bottom polycarbonate layer, causing condensation. The condensed water flows down the panel and is collected in the form of a distilled stream. Seawater is used as a feed water input to the system. The solar still system was run during the period from 23 February 2013 to 23 March 2013. Raw seawater was obtained from the Gulf, Dammam, in eastern Saudi Arabia (26°26′24.19″N, 50°10′20.38″E). The initial concentrations of the total dissolved solids (TDS), pH, density (ρ), and electrical conductivity (EC) of the raw seawater were 41.4 ppt, 8.02, 1.04 g cm^{-3}, and 66.34 mS cm^{-1}, respectively. TDS and EC were measured using a TDS-calibrated meter (Cole-Parmer Instrument Co. Ltd, Vernon Hills, USA). A pH meter (model: 3510 pH meter, Jenway, UK) was used

to measure pH. A digital-density meter (model: DMA 35$_N$, Anton Paar, USA) was used to measure ρ. The temperatures of the feed water (T_F) and brine water (T_B) were measured by using thermocouples (T-type, UK). Temperature data for feed and brine water were recorded on a data logger (model: 177-T4, Testo, Inc., UK) at 1 minute intervals. The seawater was fed to the panel using the pump described above. The residence time – the time taken for the water to pass through the panel – was approximately 20 minutes. Therefore, the flow rate of the feed water, the distilled water, and the brine water was measured every 20 minutes. Also, the total dissolved solids of feed water (TDS$_F$) and brine water (TDS$_B$) were measured every 20 minutes.

The weather data, such as air temperature (To), relative humidity (RH), wind speed (U), and solar radiation (Rs) were obtained from a weather station near the experimental site. The productivity capacity (MD) or the amount of distilled water produced by the system in a given time, was obtained by collecting and measuring the amount of water cumulatively produced over time. All of the statistical analysis and data processing were carried out using IBM's Statistical Package for the Social Sciences Statistics 21 (SPSS Inc., Chicago, IL, USA). Stepwise linear regression analysis was used to determine the effectiveness of the developed network, with constants significant at the 5% level. The experiments involved one dependent variable (the MD of the solar desalination system) and nine independent variables (Julian day, To, RH, U, Rs, TDS$_F$, TDS$_B$, T_F, and T_B).

Neural network learning algorithms

The following learning algorithms were used in this paper to train ANNs and to select the best algorithm.

Levenberg–Marquardt algorithm

The Levenberg–Marquardt (LM) algorithm approaches the Newton method and has also been utilized for ANN training. The Newton method approaches the error of the network with second-order expression, in contrast to the former category which pursues a first-order expression. LM is common in the ANN field, where it is considered as
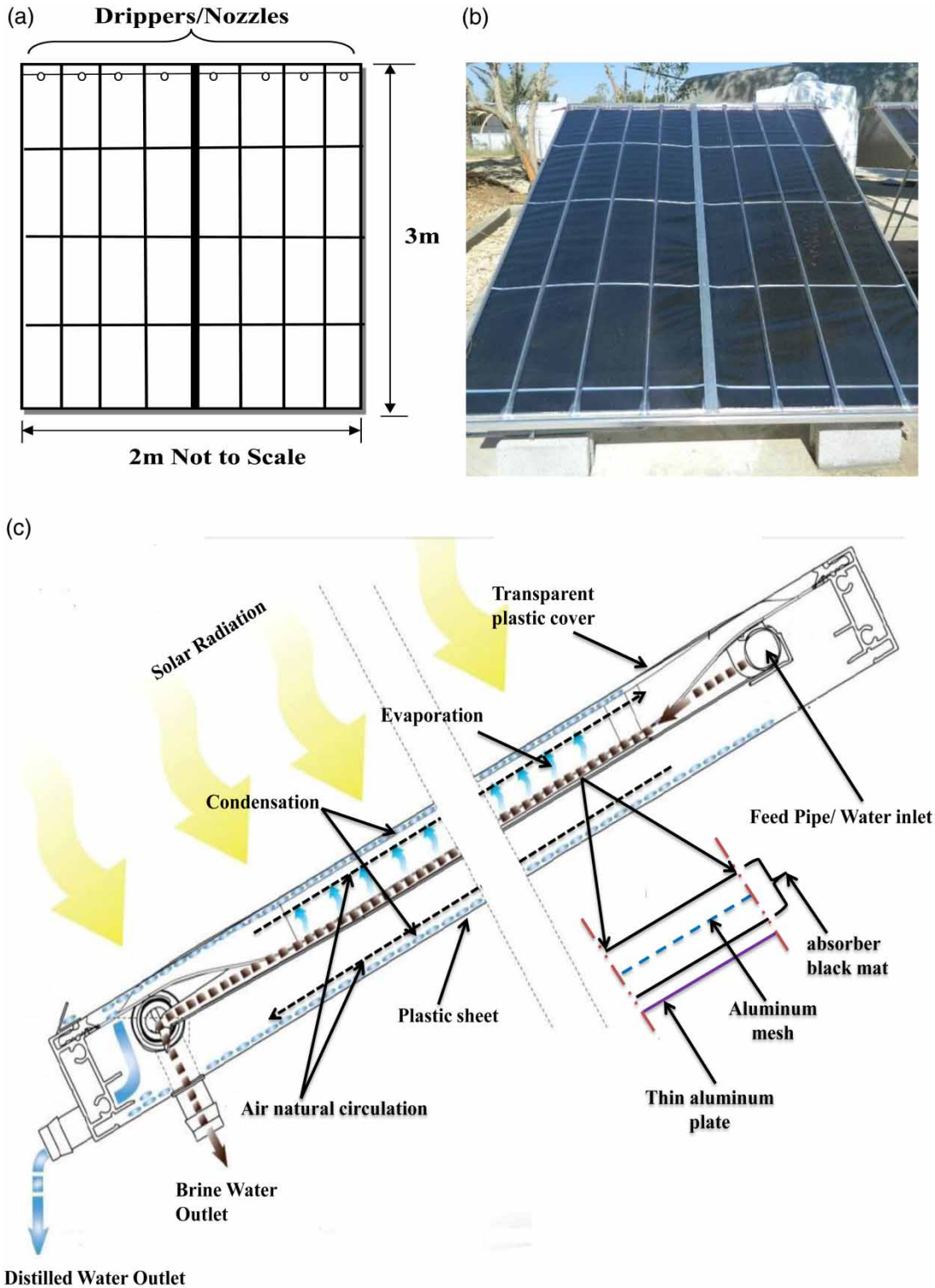
Figure 1 | Solar still panel: front section (a), picture of the panel (b), and cross-sectional view of the solar still panel (c).

the first approach for an unseen MLP training task (Hagan & Menhaj 1994).

Conjugate gradient algorithm

Conjugate gradient (CG) is a balanced algorithm. It does not need the computation of the second derivatives and confirms convergence to a local minimum of the second-order function. CG becomes approximate to a local minimum of the second-order function after a definite number of iterations. A specific type of this algorithm is expressed as conjugate gradient backpropagation with Fletcher–Reeves restarts (CGF) (Ayat et al. 2013).

Resilient backpropagation algorithm

The resilient backpropagation (RP) algorithm removes the impacts of the magnitudes of the partial derivatives (Anastasiadis et al. 2005). Only the sign of the derivative is used to determine the direction of the weight update, and the magnitude of the derivative has no impact on the weight update (Sharma & Venugopalan 2014).

Detailed information for the three algorithms mentioned above is tabulated in Table 1.

ANN application

The ANN used in this study was a feed-forward BP neural network with three layers: an input layer, one hidden layer, and an output layer. The ANN was trained in turn with the three different algorithms mentioned above. Nine variables were utilized as input parameters for the input neurons of the input layer. These variables were JD, To ($°C$), RH (%), U (km/h), Rs (W/m^2), T_F ($°C$), T_B ($°C$), TDS$_F$ (PPT), and TDS$_B$ (PPT). The output layer consisted of the neurons related to the water productivity (MD, L/m^2/h). An ANN with neuron numbers (9, 20, 1) was built, trained, tested, and validated by MATLAB 7.13.0.564 (R2011b) using neural network Tool (nntool). The available data set, which contained 160 input vectors and their corresponding output vectors from the experimental work, was divided randomly into training, testing, and validation subsets. In total, 70% of these data were used for training, 15% for testing, and 15% for validation. Table 2 shows the summary statistics of the inputs and output used for the developed ANN model. After trial and error, the number of the hidden layer was chosen as 20 in this paper.

ANN algorithm performance evaluation criteria

The ANN algorithms were evaluated by calculating four standard statistical performance evaluation criteria. These parameters were the coefficient of the root mean square error (RMSE), the efficiency coefficient (E), the overall index of model performance (OI), and the coefficient of

Table 1 │ List of algorithms

Algorithm	Training function	Computation	Description	References
LM	Trainlm	$\Delta w_k = -H'_k g_k$ $H' = J'J$ $g = J'e$	It calculates Jacobian matrix (J) including first derivatives through the BP method, which is less complex compared to the Hessian matrix (H). The network error vector is e.	Ayat et al. (2013), Talaee (2014)
CGF	Traincgf	$p_0 = -g_0$ $\Delta w_k = \alpha_k p_k$ $p_k = -g_k + \beta_k p_{k-1}$ $\beta_k = \dfrac{g'_k g_k}{g'_{k-1} g_{k-1}}$	By searching in the steepest descent direction, the iteration starts. A search line method is used to find the optimal current search direction α. Next search direction β is found such that it is conjugate to the previous search directions.	Ayat et al. (2013)
RP	Trainrp	$\Delta w_k = -\text{sign}.\left(\dfrac{\Delta E_k}{\Delta w_k}\right).\Delta_k$	Only the sign of the partial derivative is considered to indicate the direction of the weight update.	Ayat et al. (2013), Talaee (2014)

Table 2 | Descriptive statistics for the inputs and output of the developed ANN model

| Statistics | Inputs | | | | | | | | | Output |
	JD	To (°C)	RH (%)	U (km/h)	Rs (W/ m²)	T_F (°C)	T_B (°C)	TDS$_F$ (PPT)	TDS$_B$ (PPT)	MD (L/m²/h)
Mean	74.92	26.64	23.36	2.44	587.55	36.66	49.58	80.23	95.54	0.50
Std. dev.	23.22	3.68	12.90	3.12	181.93	4.27	8.16	29.42	29.59	0.24
Kurtosis	−1.04	−0.56	2.93	3.39	−0.32	−0.02	0.26	−1.32	−1.63	−1.06
Skewness	0.81	−0.40	1.93	2.03	−0.59	−0.78	0.21	0.30	−0.04	−0.08
Minimum	54	16.87	12.90	0.00	75.10	22.1	27.59	41.4	46.2	0.05
Maximum	113	33.23	70	12.65	920.69	42.35	68.69	130	132.8	0.97
Count	160	160	160	160	160	160	160	160	160	160

residual mass (CRM). They are formulated mathematically as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (x_{p,i} - x_{o,i})^2}{n}} \qquad (7)$$

$$E = 1 - \frac{\sum_{i=1}^{n} (x_{o,i} - x_{p,i})^2}{\sum_{i=1}^{n} (x_{o,i} - \bar{x}_o)^2} \qquad (8)$$

$$\text{OI} = \frac{1}{2}\left(1 - \left(\frac{\text{RMSE}}{x_{\max} - x_{\min}}\right) + E\right) \qquad (9)$$

$$\text{CRM} = \frac{\left(\sum_{i=1}^{n} x_{p,i} - \sum_{i=1}^{n} x_{o,i}\right)}{\sum_{i=1}^{n} x_{o,i}} \qquad (10)$$

where $x_{o,i}$ = observed value; $x_{p,i}$ = predicted value; n = number of observations; $x_{\max}$ = maximum observed value; $x_{\min}$ = minimum observed value; and $\bar{x}_o$ = averaged observed values.

RMSE has been used by various authors to compare predicted and measured parameters (Arbat et al. 2008). The RMSE values indicate how much the predictions under- or overestimate the measurements. Legates & McCabe (1999) stated that RMSE has the advantage of expressing the error in the same units as the variable, thus providing more information about the efficiency of the model. The lower the RMSE, the greater the accuracy of prediction. An efficiency (E) value of 1.0 implies a perfect fit between measured and predicted data and this value can be negative. The OI parameter was used to verify the performance of mathematical models. An OI value of 1 for a model denotes

a perfect fit between the measured and predicted values (Alazba et al. 2012). The CRM parameter shows the difference between measured and predicted values relative to the measured data. The CRM is used to measure the model tendency to overestimate or underestimate the measured data. A zero value points out a perfect fit, while positive and negative values indicate an under- and over-prediction by the model, respectively. The best training algorithm was chosen on the basis of the smallest RMSE and CRM and highest E and OI values.

RESULTS AND DISCUSSION

Experimentally, the average solar still production MD during the entire operation was 0.50 L/m²/h (approximately 5 L/m²/day). This is consistent with the findings of Kabeel et al. (2012) and Radhwan (2004).

General ANN model performance with different algorithms

Figure 2 depicts variations in the gradient error, the value of μ, and the validation checks utilizing the LM algorithm. As shown in this figure, at 6 epochs the gradient error is about 0.00053 and the number of validation checks is 6. Also, the figure shows that the training process was stopped owing to reaching the minimum gradient error at epoch 6. On the other hand, Figure 3 displays variations in the gradient error, validation checks and step size using CGF. As shown in the figure, the gradient error is about 0.011468.
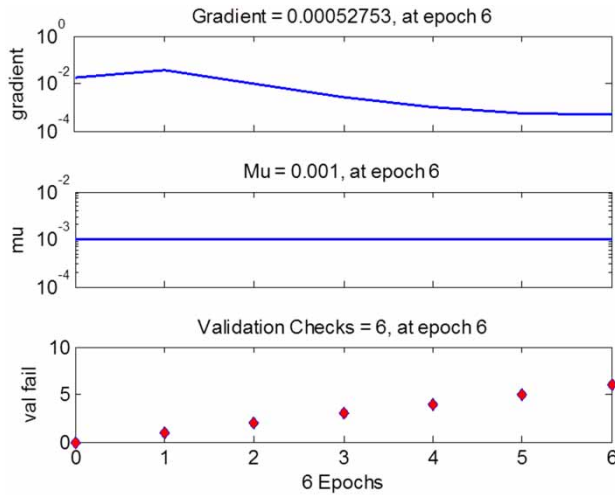
Figure 2 │ Variation of the gradient error, μ and validation error with the LM algorithm.

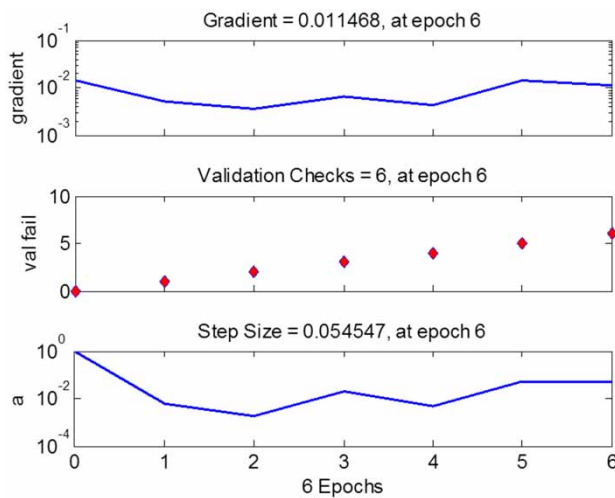**Figure 4** │ Variation of the gradient error and validation checks with the RP algorithm.

Figure 3 │ Variation of the gradient error and validation checks with the CGF algorithm.

At 6 epochs, the number of validation checks is 6 and the step size is 0.054547. Furthermore, Figure 4 shows variations in the gradient error and validation checks utilizing the RP algorithm. The figure shows that at 6 epochs, the gradient error is approximately 0.085518 and the number of validation checks is 6. Thus, the results indicate that the LM algorithm is the best, as it gives the least gradient error, followed by the CGF and lastly the RP.

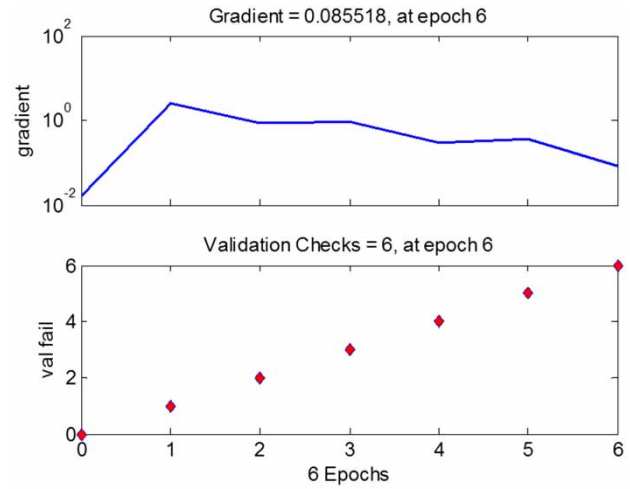In addition, Figures 5–7 depict regression analysis incorporating training, testing, validation, and all data using the LM, CGF, and RP algorithms, respectively. The dashed line in Figure 5 represents the condition in which outputs and targets are equal; data points are represented by circles. In addition, the solid line denotes the best fit between outputs and targets. The data (circles) are clustered along the dashed line, indicating that the output values are close to the target values. The correlation coefficients (R) for the training, testing, and validation processes are 0.99285, 0.99809, and 0.99651, respectively. Furthermore, the overall R value is 0.99437 for the model based on the LM algorithm. For the CGF algorithm shown in Figure 6, the values of R for the training, testing, and validation processes are 0.98843, 0.98628, and 0.9958, respectively. Moreover, the R value is 0.98941 for the total response with the CGF algorithm. For all data, the slope (m) and intercept (b) values are approximately 0.98 and 0.0088, respectively, and the fit of the model prediction is excellent. Also, in Figure 7, which shows results using the RP algorithm, the R values for the training, testing, and validation processes are 0.98406, 0.98865, and 0.99046, respectively. The overall R value is 0.9853 for the model based on the RP algorithm. The foregoing results show that the overall R value of the LM algorithm is the highest among all of the algorithms, followed by the R values of the CGF and RP algorithms. Generally, Figures 5–7 demonstrate and prove the effectiveness of the model developed for predicting solar still production, based on the use of any of the three algorithms.

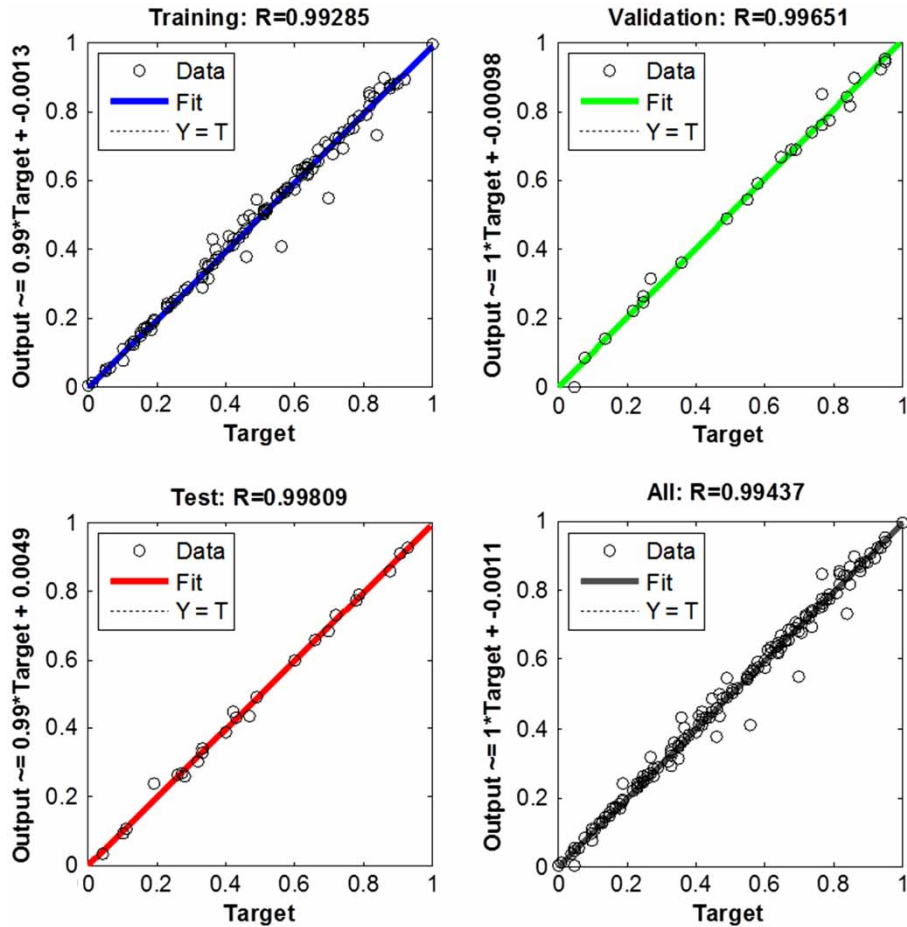A stepwise regression analysis was applied experimentally to the obtained total data for determining the

Figure 5 │ Regression analysis with training, validation, testing and all data with the LM algorithm.

effectiveness of the ANN model. Table 3 shows the degree of significance of the input parameters, as determined by a P value of <0.05. On this basis, the significance of the input variables can be ranked in the following order: Rs, TDS_F, TDS_B, and RH. Thus, it is possible to predict the water productivity of the solar still utilizing these parameters only. Moreover, it is clear to us that the R value with any algorithm is higher than the R value (0.961) with the stepwise regression analysis. This indicates and confirms that the ANN model is the most accurate compared with the stepwise regression method.

Trade-off between the learning algorithms

Table 4 displays the results of the statistical parameters, RMSE, ME, OI, and CRM, which were the numerical indicators used to assess the agreement between predicted and observed solar still productivity values with different learning algorithms throughout the modeling stages. It can be seen that the RMSE and CRM values are close to zero, whereas the E and OI values are very close to 1, demonstrating excellent agreement between observed and predicted outcomes from ANN. The very small deviation between observed and predicted outcomes confirms the effectiveness of ANN for modeling and predicting solar still production MD. The average (AVG), standard deviation (STDEV), and coefficient of variation (CV) for the RMSE of the three learning algorithms were 0.04, 0.01, and 0.27 during training, 0.03, 0.01, and 0.16 during testing, and 0.03, 0.02, and 0.67 during validation, respectively (Table 4). During training and testing, the RMSE was lowest with LM and highest with RP. However, during validation, the highest
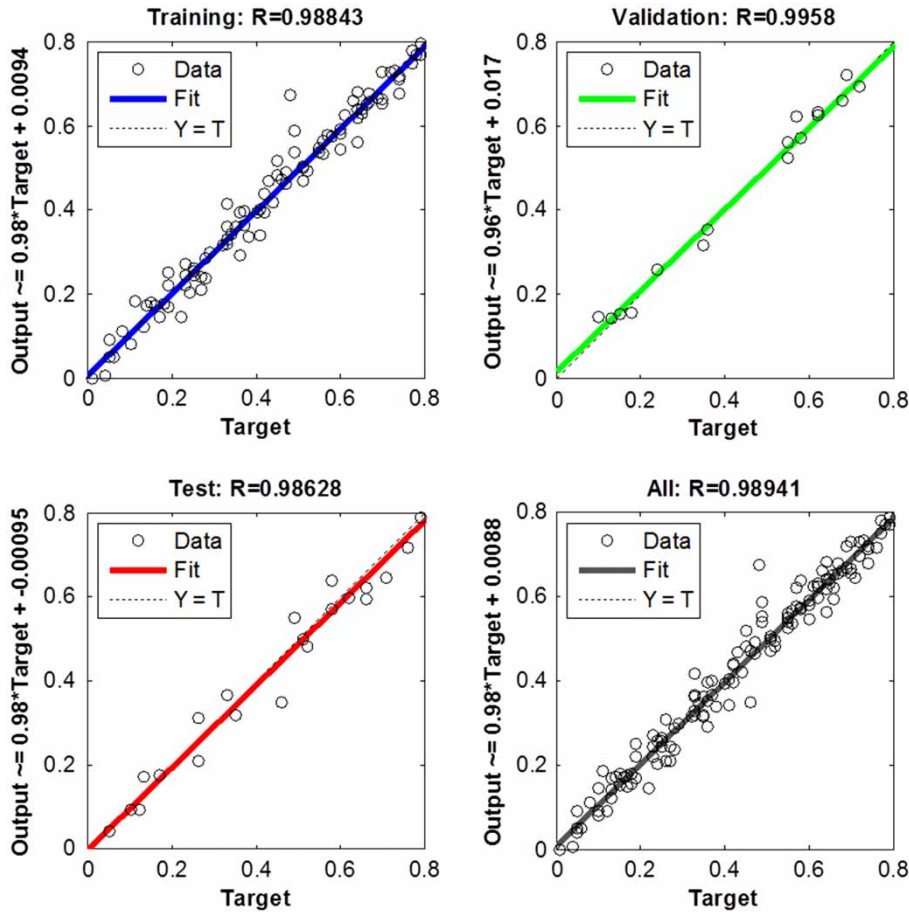
Figure 6 │ Regression analysis with training, validation, testing and all data with the CGF algorithm.

RMSE was with CGF and the lowest was with LM. At all stages, the small RMSE value with LM reflects the accuracy of the model using this algorithm. However, the differences between the three algorithms at all stages were quite small.

In addition, the AVG, STDEV, and CV for the E of the three learning algorithms were 0.98, 0.01, and 0.01 during training, 0.98, 0.002, and 0.002 during testing, 0.98, 0.002, and 0.002 during validation, respectively. The highest E value (0.987) was with the LM algorithm, whereas the lowest value (0.964) was with RP during training. Moreover, the E value was highest with the LM algorithm during validation, but during testing E was highest with CGF but very similar to the LM algorithm. Furthermore, all E values were very near to 1, as indicated in Table 4. This shows an excellent agreement between the experimental and forecasted values and proves the strength and

effectiveness of the ANN model. Similarly, the AVG, STDEV, and CV for the OI of all three learning algorithms in the training process were 0.97, 0.01, and 0.01, respectively, in testing were 0.97, 0.002, and 0.002, respectively, and in the validation process were 0.97, 0.02 and 0.02, respectively. It is noted from Table 4 that OI values with the LM algorithm were the highest in all modeling stages. Moreover, OI values with the RP algorithm were the lowest in all stages except in validation, where the CGF was the lowest. It is clear that all OI values are above 0.90 and very close to 1. This confirms that the ANN model performed very well in predicting the MD. The AVG, STDEV, and CV for the OI of all three learning algorithms in the training process were 0.01, 0.001, and 0.09, respectively, in testing were 0.01, 0.004, and 0.36, respectively, and in the validation process were 0.004, 0.004, and 1.14, respectively. It is clear from the table
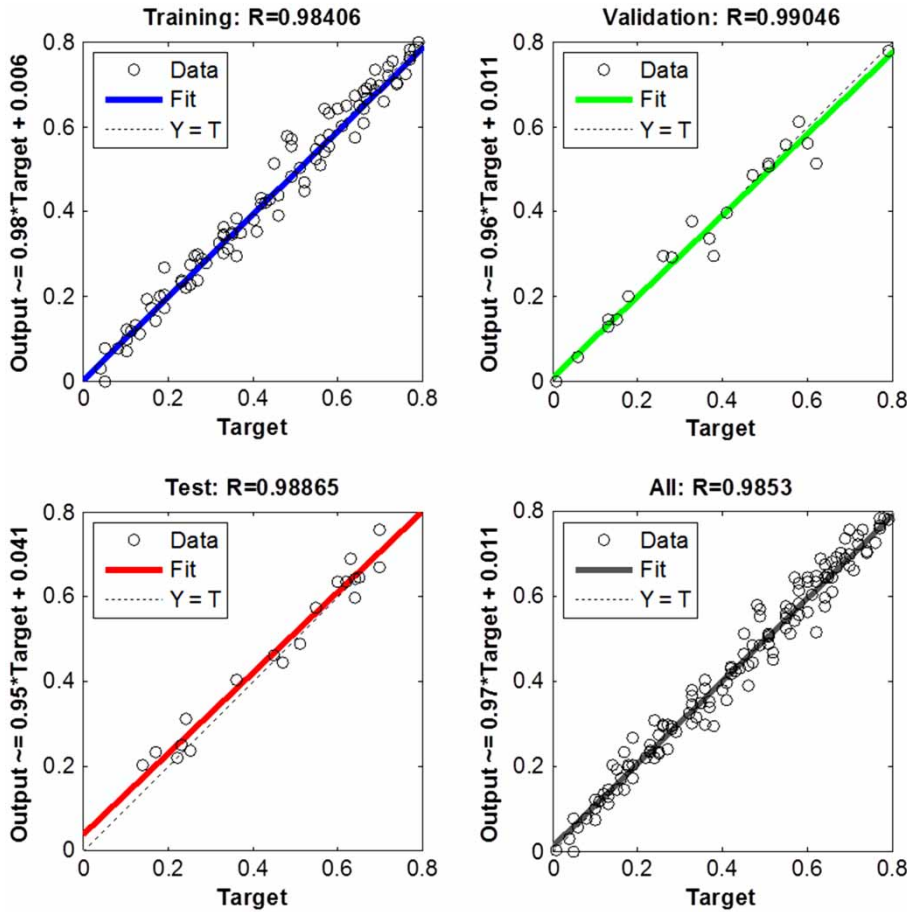
Figure 7 │ Regression analysis with training, validation, testing and all data with the RP algorithm.

Table 3 │ Results of stepwise regression analysis

| Independent parameters	Coefficient values	Std. error	t Stat	P-value	R
Constant	0.233	0.022	10.641	0.000	0.961
Rs	0.001	0.000	20.928	0.000	
TDS_F	−0.013	0.001	−15.420	0.000	
TDS_B	0.010	0.001	10.080	0.000	
RH	−0.005	0.001	−7.392	0.000	

that CRM values for all learning algorithms in the training and testing processes were below zero, which shows underestimation in predicting MD, while CRM values for LM and RP algorithms in the validation process were above zero which indicates overestimation in predicting MD. However, in both cases it is very close to zero, indicating

the accuracy of the ANN model. All values are close to each other, but the value of LM was somewhat better.

Figures 8–10 display the relative errors of predicted MD values for the training, testing, and validation data sets for the ANN model with the three learning algorithms. These figures show the differences between the predicted results of the LM, CGF, and RP algorithms based on corresponding relative errors, which will help us to identify the best performing and most accurate algorithm in forecasting MD. It is demonstrated in Figure 8 that the relative errors of MD values predicted by the LM algorithm are relatively small and the majority, almost 88%, fall in a domain approximately ranging from +10 to −10%. On the other hand, about 76.8 and 73.2% of errors fall in this range for CGF and RP, respectively, in the training stage. Furthermore, in the testing stage, it is noted from Figure 9 that around

Table 4 | Statistical parameters for evaluating the performance of the developed ANN model with the different algorithms

	LM	CGF	RP	LM	CGF	RP	LM	CGF	RP
	Training			**Testing**			**Validation**		
RMSE	0.029	0.033	0.048	0.027	0.032	0.037	0.016	0.058	0.025
E	0.987	0.982	0.964	0.983	0.985	0.981	0.996	0.954	0.991
OI	0.979	0.974	0.958	0.974	0.973	0.970	0.989	0.948	0.980
CRM	−0.007	−0.006	−0.006	−0.009	−0.014	−0.007	0.007	−0.001	0.005
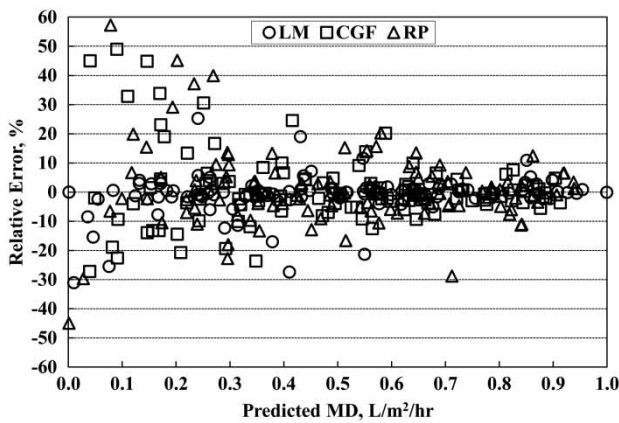
Figure 8 | Relative error for the ANN model with different learning algorithms using the training data set.
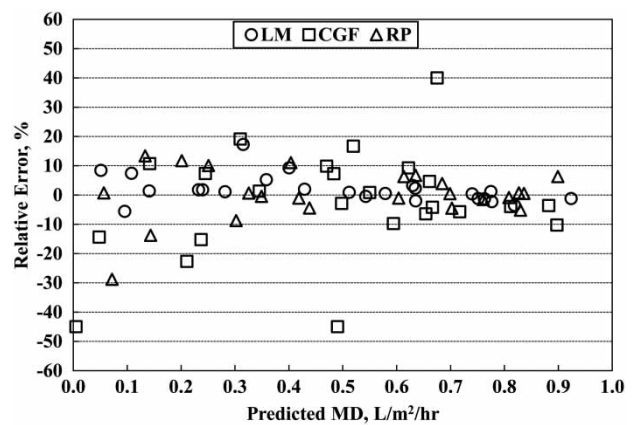
Figure 10 | Relative error for the ANN model with different learning algorithms using the validation data set.
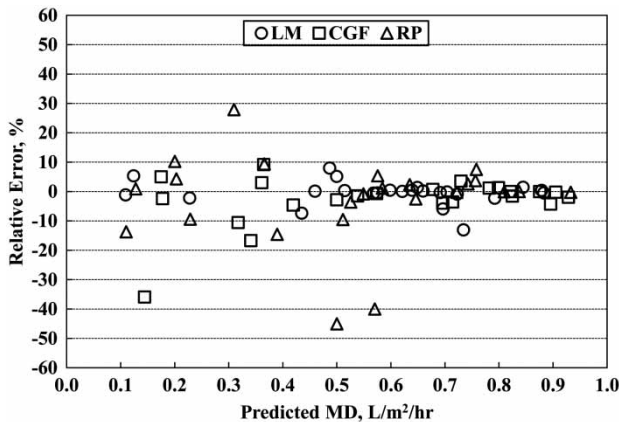
Figure 9 | Relative error for the ANN model with different learning algorithms using the testing data set.

96%, 83%, and 75% of the errors for LM, CGF, and RP algorithms, respectively, were in the vicinity of ±10%. Moreover, the relative errors for LM are very small in the validation stage compared with the CGF and RP algorithms, as shown in Figure 10. In the validation stage, we found that

96% of the errors are located in the range of ±10% for LM, while the corresponding values for each of the CGF and RP algorithms were about 58% and 75%, respectively, in the domain of ±10%. From the above it is clear that the LM algorithm is better than the CGF and RP algorithms. Generally, the range of ±10% is acceptable to judge the performance, and the values that fall outside this range were relatively few and do not affect the applicability of the algorithms in the prediction process.

CONCLUSION

This study presents an approach to forecast solar still productivity using meteorological and operational parameters based on a feed-forward BP ANN using different learning algorithms. Three learning algorithms, namely the LM algorithm, the CGF algorithm, and the RP algorithm, were adopted in training the developed ANN model.

The performance of the developed ANN model using these three algorithms was evaluated by comparing the predicted results to the experimental results using a set of standard statistical performance measures, namely RMSE, E, OI, and CRM. Comparative analysis of predicted data and real data reveals that the feed-forward BP neural network model has the ability to identify the relationship between the input and output parameters as well as the ability to predict solar still productivity to a high degree of accuracy. The LM learning algorithm with a minimum average RMSE (0.024), maximum average E (0.989), maximum average OI (0.981), and minimum average CRM (-0.003) throughout all modeling stages was found to be the most effective algorithm in predicting solar still productivity. Thus, based on these findings, the developed ANN model with the LM learning algorithm is recommended for solar still productivity prediction in a hyper-arid environment.

ACKNOWLEDGEMENT

REFERENCES

Alazba, A., Mattar, M. A., ElNesr, M. N. & Amin, M. T. 2012 Field assessment of friction head loss and friction correction factor equations. *J. Irrig. Drain. Eng. ASCE* **138** (2), 166–176.

Alsmadi, M. K. S., Omar, K. B. & Noah, S. A. 2009 Back propagation algorithm: the best algorithm among the multi-layer perceptron algorithm. *IJCSNS* **9** (4), 378–383.

Anastasiadis, A. D., Magoulas, G. D. & Vrahatis, M. N. 2005 New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing* **64**, 253–270.

Arbat, G., Puig-Bargues, J., Barragan, J., Bonany, J. & Ramirez de Cartagena, F. 2008 Monitoring soil water status for micro-irrigation management versus modeling approach. *Biosystems Eng.* **100** (2), 286–296.

Ayat, S., Farahani, H. A., Aghamohamadi, M., Alian, M., Aghamohamadi, S. & Kazemi, Z. 2013 A comparison of artificial neural networks learning algorithms in predicting tendency for suicide. *Neural Comput. Applic.* **23**, 1381–1386.

Ayoub, G. M. & Malaeb, L. 2012 Developments in solar still desalination systems: a critical review. *Crit. Rev. Env. Sci. Tec.* **42** (19), 2078–2112.

Gao, P., Zhang, L., Cheng, K. & Zhang, H. 2007 A new approach to performance analysis of a seawater desalination system by an artificial neural network. *Desalination* **205** (1–3), 147–155.

Hagan, M. T. & Menhaj, M. B. 1994 Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* **5** (6), 989–993.

Hamdan, M. A., Haj Khalil, R. A. & Abdelhafez, E. A. M. 2013 Comparison of neural network models in the estimation of the performance of solar still under Jordanian climate. *JOCET* **1** (3), 238–242.

Hamdan, M., Badri, L. & Abdelhafez, E. 2014 Modeling triple solar still production using Jordan weather data and artificial neural networks. *IJTEE* **7** (2), 87–93.

Haykin, S. 2009 *Neural Networks and Learning Machines*, 3rd edn. Pearson Education, Upper Saddle River, NJ.

Jang, J.-S. R., Sun, C.-T. & Mizutani, E. 1997 *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice-Hall, Upper Saddle River, NJ.

Kabeel, A. E., Hamed, M. H. & Omara, Z. M. 2012 Augmentation of the basin type solar still using photovoltaic powered turbulence system. *Desalin. Water Treat.* **48**, 182–190.

Kalogirou, S. A. 2001 Artificial neural networks in renewable energy systems applications: a review. *Renew. Sustain. Energy Rev.* **5** (4), 373–401.

Kalogirou, S. A., Mathioulakis, E. & Belessiotis, V. 2014 Artificial neural networks for the performance prediction of large solar systems. *Renew. Energ.* **63**, 90–97.

Khayet, M. & Cojocaru, C. 2013 Artificial neural network model for desalination by sweeping gas membrane distillation. *Desalination* **308**, 102–110.

Lee, Y. G., Lee, Y. S., Jeon, J. J., Lee, S., Yang, D. R., Kim, I. S. & Kim, J. H. 2009 Artificial neural network model for optimizing operation of a seawater reverse osmosis desalination plant. *Desalination* **247**, 180–189.

Legates, D. R. & McCabe, J. 1999 Evaluating the use of 'goodness-of fit' measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* **35** (1), 233–241.

Porrazzo, R., Cipollina, A., Galluzzo, M. & Micale, G. 2013 A neural network-based optimizing control system for a seawater-desalination solar-powered membrane distillation unit. *Comput. Chem. Eng.* **54**, 79–96.

Radhwan, A. M. 2004 Transient performance of a steeped solar still with built-in latent heat thermal energy storage. *Desalination* **171**, 61–76.

Ramírez, E., Castillo, O. & Soria, J. 2003 Hybrid system for cardiac arrhythmia classification with fuzzy k-nearest neighbors and multilayer perceptrons combined by a fuzzy inference system. In: *Proceedings of the International Joint Conference on Neural Networks*. Academic Press, Barcelona, Spain.

Rojas, R. 1996 *Neural Networks: a systematic introduction*. Springer-Verlag, Berlin, pp. 151–184.

Santos, N. I., Said, A. M., James, D. E. & Venkatesh, N. H. 2012 Modeling solar still production using local weather data and artificial neural networks. *Renew. Energ.* **40** (1), 71–79.

Sharma, B. & Venugopalan, K. 2014 Comparison of neural network training functions for hematoma classification in brain CT images. *IOSR-JCE* **16** (1), 31–35.

Talaee, P. H. 2014 Multilayer perceptron with different training algorithms for streamflow forecasting. *Neural. Comput. Appl.* **24**, 695–703.

Tripathy, P. P. & Kumar, S. 2008 Neural network approach for food temperature prediction during solar drying. *Int. J. Thermal Sci.* **48** (7), 1452–1459.