Photovoltaic Module Temperature Estimation: A Comparison Between Artificial Neural Networks and Adaptive Neuro Fuzzy Inference Systems Models

J. Tziu Dzib[1], E.J. Alejos Moo[1], A. Bassam[1](✉), Manuel Flota-Bañuelos[1], M.A. Escalante Soberanis[1,2], Luis J. Ricalde[1], and Manuel J. López-Sánchez[1]

[1] Facultad de Ingeniería, Universidad Autónoma de Yucatán,
Av. Industrias no contaminantes, Mérida, Yucatán, Mexico
josetziu@me.com, e.alejos@outlook.com, mauricio.escalante@hotmail.com,
{baali,manuel.flota,lricalde,manuel.lopez}@correo.uady.mx

[2] Clean Energy Research Centre, University of British Columbia, 2360 East Mall,
Vancouver, BC V6T 1Z3, Canada

Abstract. The main objective of this paper is to present a comparison between two models for estimation of a photovoltaic system's module temperature (T_{mod}) using Artificial Neural Networks and Adaptive Neuro Fuzzy Inference Systems. Both estimations use measurements of common operation variables: current, voltage and duty cycle (d) from a power converter of the photovoltaic system as input variables and T_{mod} as a desired output. The models used the same database for the training process, different training strategies were evaluated with the objective to find which model has the best estimation with respect to the T_{mod}. Subsequently, the output results from these architectures are validated via the Root Mean Squared Error, Mean Absolute Percentage Error and correlation coefficient. Results show that the Artificial Neural Network model in comparison with Adaptive Neuro Fuzzy Inference System model provides a better estimation of T_{mod} with $R = 0.8167$. Developed models may have an application with smart sensors on cooling systems for photovoltaic modules with the objective of improving their operation efficiency.

Keywords: Renewable energy · Artificial intelligence · Photovoltaic system · Module temperature · Levenberg-Marquardt algorithm · Statistical comparison

1 Introduction

The energy of Sun is the most abundant energy source on planet earth, it is renewable and available for direct or indirect use, i.e. solar radiation, wind, biomass, thermal, etc. If only 0.1 % of the solar energy that reaches the earth could be turned into electrical energy at an efficiency of 10 %, there would be 4 times

A. Martin-Gonzalez and V. Uc-Cetina (Eds.): ISICS 2016, CCIS 597, pp. 46–60, 2016.
DOI: 10.1007/978-3-319-30447-2_4

more energy available than the world production capacity (5000 GW) [1]. One way to harvest this energy source is through the use of photovoltaic (PV) technology. Over the last decade, PV technology has had a rapid increase in usage compared to other types of renewable energy sources [2]. A Photovoltaic module (PVM) converts solar radiation into Direct Current (DC) which is transferred to a power condition unit [2], this means that high levels of radiation improve the overall output of the PVM, but this has a side effect. Higher levels of radiation mean that the flux of photons moving within a PV cell is also higher, which results in an increase in temperature of the PVM [3]. However, the rise in temperature within the module reduces its efficiency, thus producing low levels of voltage and current [4], this causes a problem since it is necessary to extract as much energy as possible from the system in order to make it effective. The implementation of cooling systems offers a solution to this problem, but these systems require exact measurements of T_{mod} for their optimal operation [5]. However, temperature sensors are usually imprecise, require maintenance and are sensible to climate conditions such as ambient temperature, wind speed, radiation flux and thermal properties of the materials of the PVM, making them unreliable when used on these systems [6]. Estimation methods offer an alternative to temperature sensors, although, the unpredictability and the non-linear behavior of the temperature tends to be a problem when trying to estimate it. Artificial Intelligence (AI) techniques have recently had multiple applications on engineering in general and this is due to the fact that they provide a better solution as these often do not need statistic data and solve problems more complex than their own programming at higher speeds [7]. AI covers multiple techniques such as Artificial Neural Networks (ANN), Adaptive Neuro Fuzzy Inference Systems (ANFIS), amongs others.

Different publications related to application of ANN and ANFIS on PV technology can be found elsewhere. García-Domingo et al. [8] proposed an electric characterization of a concentrating PV using ANN. Paul et al. [9] presented an ANN model to identify and optimize statistics representing insulation availability by a solar PV system. Mellit et al. [10] estimated the power produced of a photovoltaic module with an ANN estimation model. Salah and Ouali [11] proposed two methods of maximum power point tracking using ANN and ANFIS controllers for PV systems. Salaiman [2] presented the modeling of operating PV module temperature using ANN with solar irradiance and ambient temperature as inputs for the ANN architecture.

The aim of this investigation is to design ANN and ANFIS models to estimate the T_{mod} of a PV system and compare these two models to determinate which is the best for estimation of this variable. This paper is organized as follows: an overview on PV experimental systems is proposed in the second section, artificial neural networks and adaptive neuro fuzzy inference system models are described in the third section, the fourth section is devoted to the training, results and comparison process. Finally, conclusions are presented.

2 Material and Method

2.1 Photovoltaic Experimental System

A photovoltaic setup was developed and installed in order to acquire experimental data and evaluate its performance, the PV system was installed in Mérida, Yucatán, México (20°56'18.2"N 89°36'55.8" W), a schematic diagram of this setup is illustrated in Fig. 1. The system consists of a solar PVM, a current sensor with a 0 to 20/40/80 ADC selector and 0–5 VDC output, a voltage sensor, a miniature infrared temperature sensor, a data acquisition (DAQ) USB Device, and finally, a laptop with NI *LabView*$^{\mathrm{TM}}$ software, see Table 1 for component models.

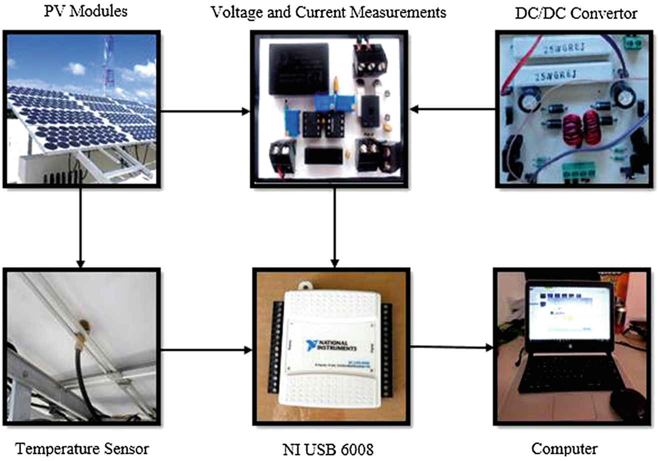


Fig. 1. Schematic diagram for experimental system setup.

The measured T_{mod} was taken by the infrared temperature sensor, the voltage and current of the PVM were measured with the voltage and current sensors, respectively with a duty cycle d set by the DC/DC converter. Finally, this data is collected in a synchronized pattern using the DAQ USB Device and then dispatched to the computer where it is analyzed via *LabView*$^{\mathrm{TM}}$, this software provides a user-friendly interface and allows the setting of sampling time intervals.

The database consists of registers taken every 10 s, at 20 min an average value of the registers is calculated in order to obtain a representative sample and the result is moved into the database. Sample consisting of a total of 1045 data pairs with significant temperature variations (see Fig. 5) was selected for training and validation purposes of the AI models, this sample presents PVM parameters under different climate conditions. Table 2 illustrates a list of input and output variables used for the ANN and ANFIS models; Figs. 2, 3 and 4 represent

Table 1. Photovoltaic experimental systems characteristics.

Component	Model
Photovoltaic module	YL110Wp
Current sensor	H970LCA
Voltage sensor	MCR-VDC-UI-B-DC
Temperature sensor	PM-HA-21-MT-CB
DAQ device	NI USB-6008/6009
Laptop	1.8 GHz i7 8DB DDR3 RAM

graphical behavior of Current (I), Voltage (V) and Duty Cycle (d) respectively as input variables, and Fig. 5 represent the behavior of module temperature as output variable.

Table 2. Characteristics of input and output variables about ANN and ANFIS models.

Parameters	Samples	Min.	Max.	Unit
Input				
d	1045	20	60	-
Voltage	1045	0.0048	33.2325	V
Current	1045	4.8428e-04	3.3232	A
Output				
Temperature	1045	289.6500	312.5389	K

2.2 Artificial Neural Network

ANN is an interconnected set of processing units that uses mathematical and computational techniques to solve problems from complicated, imprecise or missing data [12]. Each of these units is called a perceptron or neuron and has an incoming weight, bias and an output given by the transfer function of the sum of the inputs, see Fig. 6. The function of the output neuron can be mathematically expressed as:

$$u(x,w) = \sum_{i=1}^{n} f(w_i x_i + b), \tag{1}$$

where $u(x,w)$ is the output of the neuron, w_i is the synaptic weights, x_i is the input data and b is the bias value.

An ANN is generally organized on three layers: Input, hidden and output layer [13]. The ANN training can be divided in two phases: The first phase consists of updating the neuron activation values with a chosen learning algorithm,

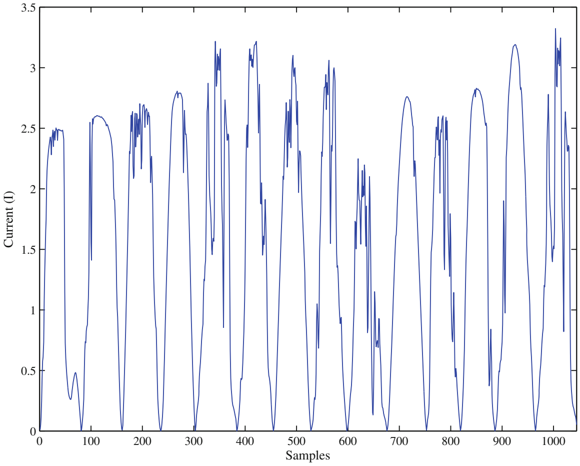


Fig. 2. Graphical representation of the current.

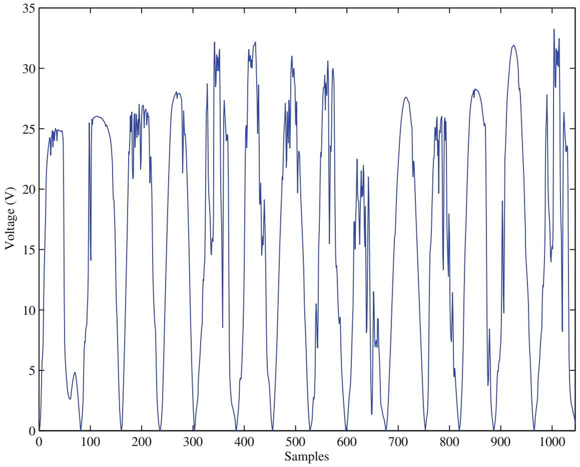


Fig. 3. Graphical representation of the voltage.

the second phase updates weights to minimize the function error measuring the difference between the desired and actual output [14].

Developing an ANN requires selection of the optimal training architecture, often set using information given by the experience and knowledge of the user [2].

2.3 Adaptive Neuro Fuzzy Inference System

ANFIS is a multilayer network that uses neural network learning algorithms and fuzzy logic to map an input space to an output space. There are two types of fuzzy inference systems (FIS): Mamdani [15] and Sugeno [16]; Mamdani being more

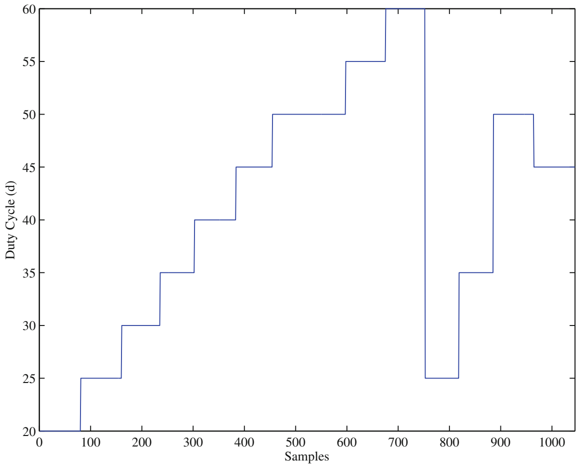


Fig. 4. Graphical representation of the duty cycle.

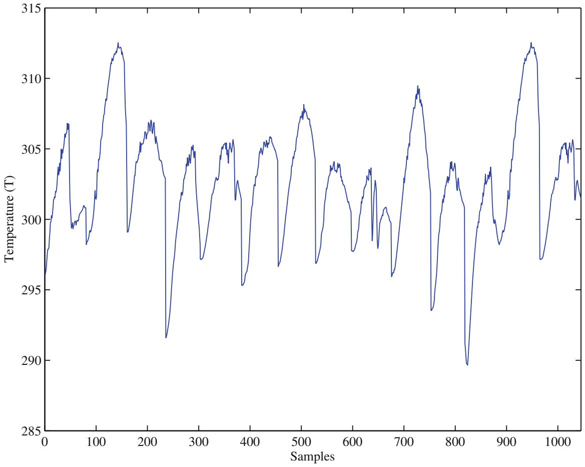


Fig. 5. Graphical representation of the module temperature.

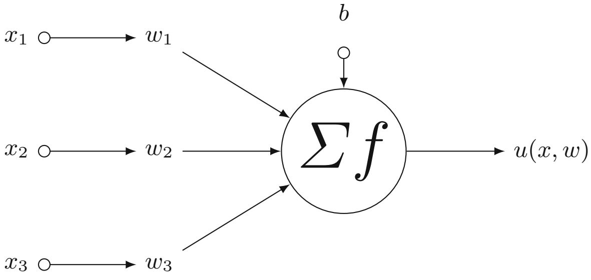


Fig. 6. A typical elementary network with 3 inputs.

intuitive and suited to human input, whereas Sugeno is more computationally efficient and works well with optimization and adaptive techniques. The consequence parameter in Sugeno FIS can be either a linear equation or a constant coefficient. The linear equation called "first-order Sugeno FIS" and the constant coefficient called "zero-order Sugeno FIS" are proposed by Jang [18]. Given the advantages of the Sugeno FIS, this model is used in this study, see Fig. 7.

Five layers are used to construct this system. Each layer consists of n number of nodes described by their function. Nodes denoted by squares are called "adaptive nodes", these represent parameter sets that are modifiable; nodes denoted by circles are called "fixed nodes", these fixed parameters set in the system. The output data from the nodes in a layer will be the input data of the next layer.

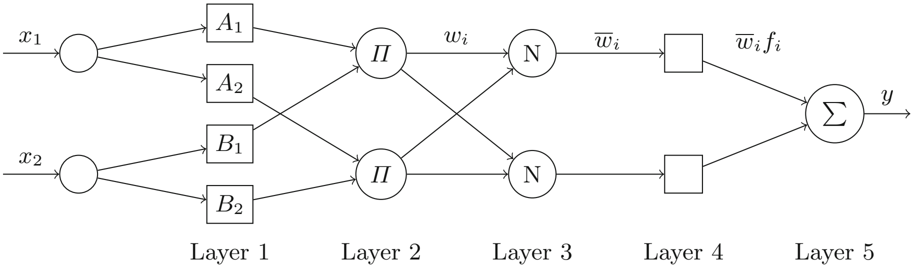


Fig. 7. Simplified ANFIS architecture.

To demonstrate the procedure of the ANFIS, a simple architecture is proposed. The system in Fig. 7 consists of two inputs, x_1 and x_2, and one output, y. Suppose the system is a first-order Sugeno FIS with a rule base contaning two fuzzy if-then rules expressed as:

Rule 1:
If x_1 is A_1 and x_2 is B_1,
then $f_1 = p_1x_1 + q_1x_2 + r_1$.
Rule 2:
If x_1 is A_2 and x_2 is B_2,
then $f_2 = p_2x_1 + q_2x_2 + r_2$.

where p_i, q_i and r_i (i = 1, 2) are the linear parameters of the consequent part of the Sugeno FIS. Each layer of the model is as follows (note that O_i^j denotes the output of the i-th node and the j-th layer):

Layer 1: Input nodes. Each node in this layer generates membership grades for each input. For instance, the function of the i-th may be a Gaussian MF:

$$O_i^1 = \mu A_i(x) = \ e^{\frac{-(x_i-b_i)^2}{2a_i^2}}, i = 1, 2. \tag{2}$$

where x is the input to node i, A_i is the MF associated with this node and a_i, b_i are the parameters set that change the shape of the MF. Parameters in this layer are called *premise parameters.*

Layer 2: Rule nodes. Each node in this layer calculates the firing strength (output) of a rule via multiplication.

$$O_i^2 = w_i = \mu A_i(x_1)\mu B_i(x_2), i = 1, 2. \tag{3}$$

In ANFIS the total number of rules is given by Eq. (4)

$$R_n = j^i, \tag{4}$$

where i is the number of inputs, and j is the number of MFs per input.

Layer 3: Average nodes. Each node in this layer calculates the ratio of the i-th rule's firing strength to the total of all firing strengths:

$$O_i^3 = \overline{w}_i = \frac{w_i}{\sum_i w_i}, i = 1, 2. \tag{5}$$

Layer 4: Consequent nodes. Each node in this layer computes the contribution the i-th rule towards the overall output with the function:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x_1 + q_i x_2 + r_i), i = 1, 2. \tag{6}$$

where $\overline{w}_i$ is the output of the layer 3, and p_i, q_i, r_i are the parameter sets. Parameters in this layer are called *consequent parameters.*

Layer 5: Output node. The single node in this layer computes the overall output as the sum of all contribution from each rule:

$$O_i^5 = y_1 = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{7}$$

2.4 Statistical Criteria

For training, validation and comparison processes for ANN and ANFIS models, a statistical analysis is performed and applied using the following statistical test parameters: Correlation Coefficient (R), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), see Table 3. R provides information on the linear relationship between the measured and estimated values. RMSE parameter is a frequently-used measure of the differences between values predicted by a model and the actual values observed. MAPE parameter is the absolute computed average of errors (%) by which estimated predictions of a variable differ from their actual values. The knowledge of this statistical parameter aids to evaluate whether the estimated predictions are underestimated or overestimated with respect to actual or expected data [19].

Where T$_{mod}$ is the measured temperature and T$_{sim}$ is the simulated temperature.

Table 3. Statistical criteria used for evaluation.

Statistical parameters	Equation
Correlation Coefficient (R)	$R = \frac{\sum_{i=1}^{N}(T_{mod}-\overline{T_{mod}})(T_{sim}-\overline{T_{sim}})}{\sqrt{\sum_{i=1}^{N}(T_{mod}-\overline{T_{mod}})^2(T_{sim}-\overline{T_{sim}})^2}}$
Root Mean Square Error (RMSE)	$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_{mod}-T_{sim})^2}$
Mean Absolute Percentage Error (MAPE)	$MAPE = \frac{1}{N}\sum_{i=1}^{N}\lvert\frac{T_{mod}-T_{sim}}{T_{mod}}\rvert$

3 Results and Discussion

3.1 Artificial Neural Network

Measurements of voltage and current from the PVM and the d factor from the DC/DC converter were selected as input variables for the ANN architecture, and T_{mod} of the PVM as the desired output. The number of neurons and transfer functions in the hidden layer must be adjusted to minimize the differences between the target and simulated output. *MatLab*™'s neural network tool was used to train and estimate the measured data, with a total of 1045 data pairs used in this model, 80 % for training and 20 % for testing and validation. All results reached for the ANN architecture were trained with 1000 iterations of 1000 epochs.

The process to determine the learning algorithm, number of neurons in the hidden layer, and activation functions is frequently set using heuristic method. In this work, eight back-propagation algorithms were studied to determine the best

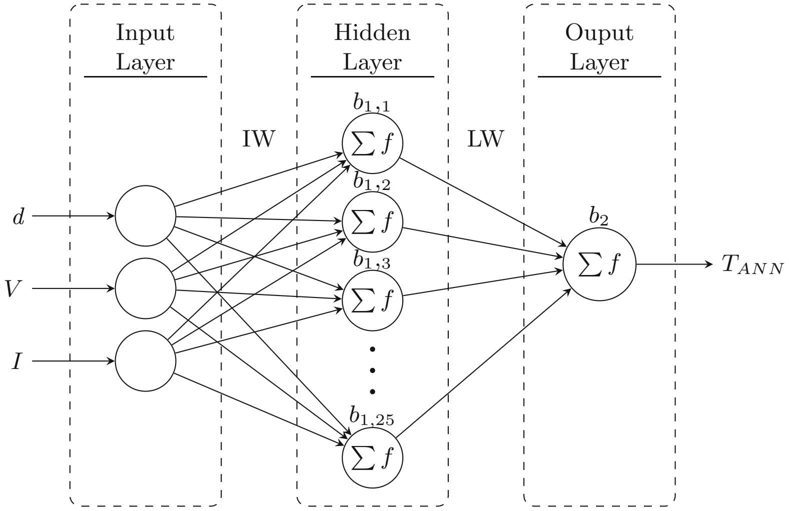


Fig. 8. Optimal ANN architecture reached.

Table 4. Comparison of back-propagation algorithms.

Back-propagation Algorithm	Mean Time (s)	RMSE	MAPE	R	Best linear equation
Levenberg-Marquardt	**2.21**	**2.4368**	**0.6413**	**0.8167**	**y = 0.67x + 99**
Bayesian regularization	19.12	2.4471	0.7753	0.8120	y = 0.62x + 91
Powell Beagle c. g.[a]	4.05	2.4827	0.8027	0.7801	y = 0.58x + 97
Batch gradient descent	13.16	2.6660	0.8851	0.6979	y = 0.58x + 108
One step secant	3.42	2.4418	0.8800	0.7969	y = 0.59x + 95
Batch gradient descent [b]	0.5	34.1916	8.6642	0.5049	y = 3.97x + 913
PolakRibiere [a]	2.44	3.1280	0.8014	0.6607	y = 0.44x + 169
Scaled [a]	1.00	3.1172	0.7930	0.6637	y = 0.44x + 168

[a] Conjugent gradient. [b] With momentum

T$_{mod}$ estimation. Table 4 shows different back-propagation algorithms trained with 25 neurons in the hidden layer. Results show that the performance between Levenberg-Marquardt (RMSE = 2.4368) and Bayesian regularization (RMSE = 2.4471) algorithms are similar but differ in the mean time of convergence (2.21 and 19.12 s respectively), Levenberg-Marquardt was over eleven times faster than the Bayesian regularization. The best prediction was found with the Levenberg-Marquardt algorithm, this algorithm performs at RMSE=2.4368 with better linear fitting ($y = 0.67x + 99$) and an execution time of 2.21 s, this is due to the LM algorithm being designed to approach second order training speed without having to compute the Hessian matrix [20].

In order to find the most efficient transfer function, two different pairs of the transfer functions (Tansig-Purelin and Logsig-Purelin) were tested for the hidden and output layer respectively, varying the number of neurons in the hidden layer and training with LM algorithm. Logsig-Purelin were the functions with the best performance. A structure 3–25-1 presents a smaller RMSE (2.4368) and greater R (0.8167) than the values trained with a combination of Tansig-Purelin transfer function.

The optimum ANN architecture was found using an evaluation with different combinations of neurons. Table 5 illustrates the statistical comparison for different ANN architectures, the finest calculation is achieved by the ANN model with 25 neurons in the hidden layer, see Fig. 8. According to the results obtained about the RMSE, MAPE and R; values for training and testing are 2.4368, 0.6413 and 0.8167, respectively.

3.2 Adaptive Neuro Fuzzy Inference System

The ANFIS model (see Fig. 9) used in this study has three inputs (V, I, d), with five membership functions assigned to each input variable, which results in having 125 total rules according to Eq. (4). The input database (containing 1045 data

Table 5. Tests with different ANN architectures.

ANN architecture	No. neurons	RMSE	MAPE	R	Best linear equation
3-01-1	1	2.4934	0.7205	0.6364	y = 0.38x + 121
3-05-1	5	2.4890	0.6586	0.7835	y = 0.58x + 96
3-10-1	10	2.4925	0.6438	0.8100	y = 0.62x + 91
3-15-1	15	2.6713	0.6430	0.8131	y = 0.62x + 90
3-20-1	20	2.4514	0.6422	0.8149	y = 0.64x + 90
3-25-1	**25**	**2.4368**	**0.6413**	**0.8167**	**y = 0.67x + 99**
3-30-1	30	2.4494	0.6436	0.8153	y = 0.64 × 90

pairs) was randomly divided into learning and testing (80 % and 20 % respectively), obtaining good representation of the data distribution and to improve the overall training process. Several MF types were tested, including triangular, trapezoidal, generalized bell, Gaussian, sigmoidal and Pi; with 100 epoch in each training session using a hybrid learning algorithm, which uses a combination of the least-squares and back-propagation gradient descent methods to model a training data set [17]. Optimum parameters were found when checking data reached minimum RMSE.

Table 6. ANFIS performance with different types of membership functions.

Function	#MF	RMSE	MAPE	R	Best linear equation
Trimf	5	3.1623	0.7679	0.6574	y = 0.49x + 153
Trapmf	5	3.2504	0.7934	0.6345	y = 0.47x + 159
Gbellmf	5	3.1821	0.7866	0.6532	y = 0.49x + 153
Gaussmf	**5**	**2.5235**	**0.6566**	**0.7996**	**y = 0.64x + 108**
Gauss2mf	5	3.1258	0.7763	0.6634	y = 0.48x + 157
Pimf	5	3.4040	0.8013	0.6132	y = 0.50x + 150
Dsigmf	5	3.1239	0.7763	0.6638	y = 0.48x + 156
Psigmf	5	3.1239	0.7763	0.6638	y = 0.48x + 156

Table 6 illustrates the ANFIS performance with different types of MF, it can be observed that the best architecture was obtained with the FIS composed by Gaussian membership function with smaller RMSE = 2.5235, MAPE = 0.6566 and higher R = 0.7996.

3.3 Comparison of ANN and ANFIS Models

The estimation capability of the ANN and ANFIS models were individually evaluated by a linear regression analysis ($y = a + bx$) between the estimated

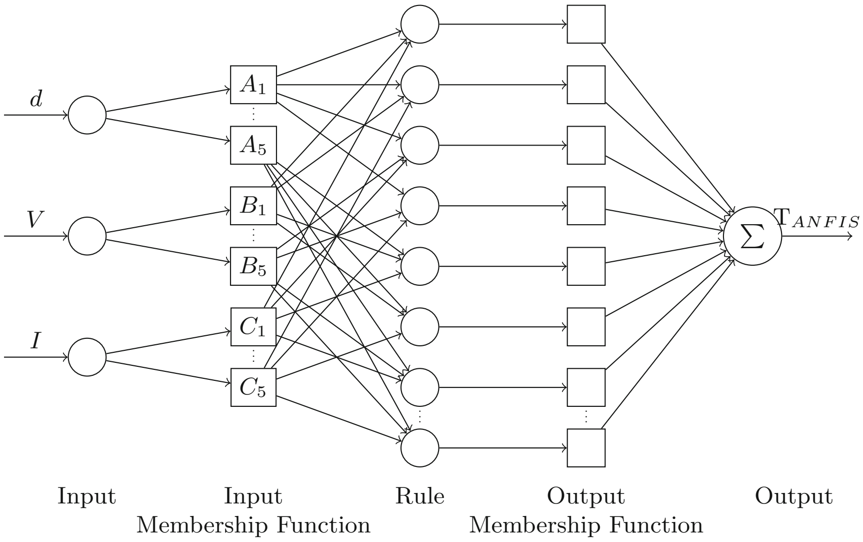
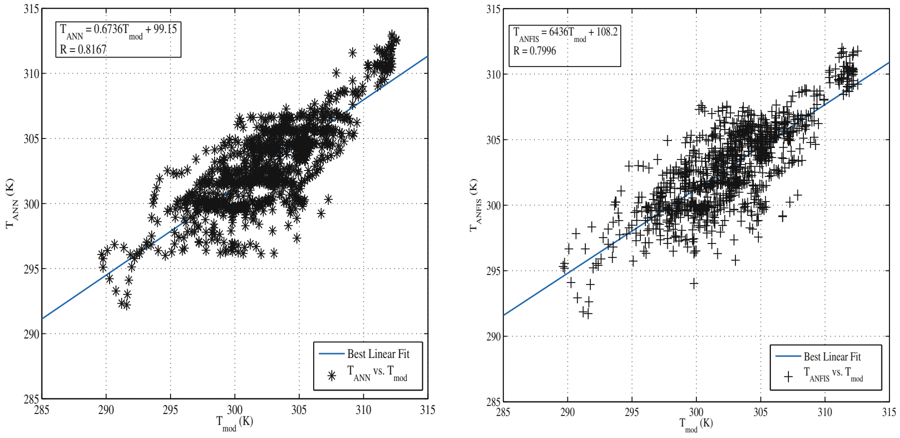


Fig. 9. ANFIS architecture used in this study. Input to Layer 3 connections are not shown.



(a) T_{mod} vs. T_{ANN}

(b) T_{mod} vs. T_{ANFIS}

Fig. 10. Comparison between error of the T_{mod}, ANN and ANFIS outputs respectively.

(T_{ANN} and T_{ANFIS} for ANN and ANFIS respectively) and measured (T_{mod}) data (using the correlation coefficient: R; the intercept: a; and the slope: b) under the same conditions [21]. Results obtained for ANN and ANFIS models are graphically shown in Fig. 10 (a, b).

The best linear regression equation for the ANN model was given by statistical parameters: a = 0.67 and b = 99 with R = 0.8120; whereas for ANFIS:

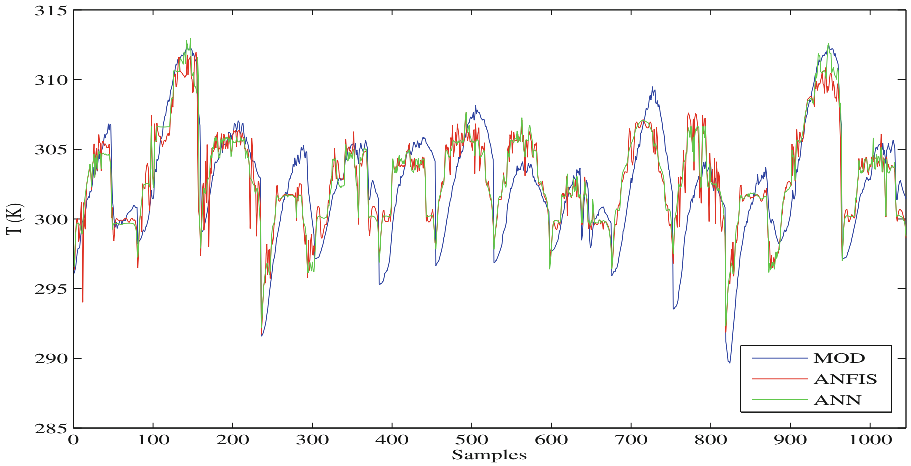


Fig. 11. Comparison between measured temperatures, ANN and ANFIS outputs.

a = 0.64 and b = 108 with R = 0.7996. According to these statistical analysis the ANN model estimation proved to be better than ANFIS for the T_{mod} approximation, although, the difference between ANN and ANFIS models is not outstanding.

With the purpose to illustrate the behavior of the estimated T_{mod} of the ANN and ANFIS in comparison with the measured data, Fig. 11 presents samples of ANN and ANFIS models estimations of this variable. It can be observed that ANN and ANFIS following the periodic behavior of the T_{mod} with ANN having better precision than ANFIS.

4 Conclusion

Application and comparison of ANN and ANFIS models for estimation of photovoltaic module temperature were investigated. Models with different functions were designed and trained by ANN and ANFIS methods. Values R, RMSE and MAPE were obtained for the ANN and ANFIS models. Comparing the performance of both models, the ANN model with Levenberg-Marquardt function had better performance in photovoltaic module temperature estimation and was selected as the best fitting model. It is also important to recognize that the prediction capability of the ANN and ANFIS could be significantly improved by an appropriate training with a larger number of field measurements under such conditions and the complex behavior of the module temperature. ANN and ANFIS could constitute useful and practical tools for the implementation of smart sensors that estimate the module temperature on a photovoltaic system. One of the applications of these smart sensors focuses in the employment of cooling systems to improve the operation performance of photovoltaic modules and increase their efficiency.

References

1. World Energy Council (WEC): Survey of Energy Resources 2013. WEC London (2013)
2. Sulaiman, S.I., Shah Alam, N.Z., Zainol, Z. Othman: Cuckoo search for determining Artificial Neural Network training parameters in modeling operating photovoltaic module temperature. In: Modelling, Identification & Control (ICMIC), pp. 306–309 (2014)
3. Butay, D.F., Miller, M.T.: Maximum peak power tracker: a solar application. Worcester Polytechnic Institute (2008)
4. Skoplaki, E., Palyvos, J.A.: On the temperature dependence of photovoltaic module electrical performance: a review of efficiency/power correlations. Sol. Energy **83**, 614–624 (2009)
5. Sherkar, G., Akkewar, A.: The cooling system of photovoltaic module and their effective efficiency. Int. J. Adv. Technol. Eng. Sci. **03**, 483–492 (2015)
6. King, D.L., Kratochvil, J.A., Boyson, W.E.: Temperature coefficients for PV modules and arrays: measurement methods. difficulties and results. In: Proceedings of the 26th IEEE Photovoltaics Specialists Conference, Anaheim, CA, USA, pp. 1183–1186 (1997)
7. Mellit, A., Benghanem, M.: Artificial neural network model for prediction solar ra-diation data: application for sizing stand-alone photovoltaic power system. In: Power Engineering Society General Meeting, Vol. 1, pp. 40–44 (2005)
8. García-Domingo, B., Piliougine, M., Elizondo, D., Aguilera, J.: CPV module electric characterisation by artificial neural networks. Renew. Energy **78**, 173–181 (2015)
9. Paul, D., Mandal, S.N., Mukherjee, D., Chaudhuri, S.R.B.: Artificial neural network modeling for efficient photovoltaic system design. In: Advanced Computer Theory and Engineering, pp. 50–56 (2008)
10. Mellit, A., Saglam, S., Kalogirou, S.: Artificial neural network-based model for estimating the produced power of a photovoltaic module. Renew. Energy **60**, 71–78 (2013)
11. Salah, C.B., Ouali, M.: Comparison of fuzzy logic and neural network in maximum power point tracker for PV systems. Electr. Power Syst. Res. **81**, 43–50 (2011)
12. Di Vi, M.C., Infield, D.: Artificial neural network for real time modelling of photovoltaic system under partial shading. In: Sustainable Energy Technologies, pp. 1–5 (2010)
13. Hasan, H., Bal, H.: Comparing performances of backpropagation and genetic algorithms in the data classification. Expert Syst. Appl. **38**, 3703–3709 (2011)
14. Malluhi, Q.M., Bayoumi, M.A., Rao, T.R.N.: An application-specific array architecture for feedforward with backpropagation ANNs. In: Application-Specific Array Processors, pp. 333–344 (1993)
15. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Mach. Stud. **7**(1), 1–13 (1975)
16. Sugeno, M.: Industrial Applications of Fuzzy Control. Elsevier Science Pub. Co., Amsterdam (1985)
17. Jang, J.S.R.: Fuzzy modeling using generalized neural networks and kalman filter algorithm. In: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI 1991), pp. 762–767 (1991)
18. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing, p. 607. Prentice Hall, New York (1997)

19. Bassam, A., Álvarez del Castillo, A., García-Valladares, O., Santoyo, E.: Determination of pressure drops in flowing geothermal wells by using artificial neural networks and wellbore simulation tools. Appl. Thermal Eng. **75**, 1217–1228 (2015)
20. Hagan, M.T., Menhaj, M.B.: Training feedforward networks with the marquardt algorithm. IEEE Trans. Neural Netw. **5**, 989–993 (1994)
21. Vieira, J., Dias, F.M., Mota, A.: Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study. Eng. Appl. Artif. Intell. **17**, 265–273 (2004)