Evaporation Modelling by using Artificial Neural Network and Multiple Linear Regression Technique

Javed Ali, G.B. Pant University of Agriculture & Technology, Pantnagar, India
Sakshi Saraf
Evaporation Modelling by using Artificial Neural Network and Multiple Linear Regression Technique

Javed Ali† and Sakshi Saraf‡

†College of Technology, G.B. Pant University of Agriculture & Technology, Pantnagar, Uttarakhand, India
‡School of Water Resources, Indian Institute of Technology, Kharagpur, West Bengal, India

†Corresponding author: javedali28@gmail.com

Received 28 September 2015; accepted 18 October 2015

Abstract
In this study an artificial neural network based evaporation estimation model was developed and evaluated the performance of the developed model for Udaipur, Rajasthan, India. Multiple linear regressions were used to estimate the pan evaporation for the study area and to model the linear correlations between a single dependent variable Y and two or more independent variables. Performance of the models was evaluated by using qualitative and quantitative indices, viz. correlation coefficient (CC) and root mean square error (RMSE). The values of root mean square error were 0.836 and 0.882 and the values of correlation coefficient were 0.970 and 0.960 for network 4-6-6-1 for training and testing period respectively. The values of root mean square error were 1.028 and 1.106 and the values of correlation coefficient were 0.941 and 0.930 for MLR model for training and testing period respectively. In this study, it was found that the evaporation estimation done through ANN was better than compared to that estimation through MLR.

Key words: Evaporation estimation; artificial neural network; multiple linear regression; meteorological data.

1. Introduction
Evaporation, as a major component of the hydrologic cycle, is important in water resources development and management since it affects the yield of river basins, the capacity of reservoirs, the consumptive use of water by crops and the yield of underground supplies. In many parts of the world, where availability of water resources is scarce, the estimation of evaporation loss is very important in the planning and management of irrigation practices, and these losses should be considered in the design of various water resources and irrigation systems (Tabari et al. 2009).

Gupta (1992) pointed out that relative humidity, wind velocity, and temperature of water and atmosphere are the climatic factors evaporation appallingly depends on. It concluded by most researchers that solar radiation, wind velocity, relative humidity, and air temperature are the most influencing factors.

Precise estimation of evaporation is crucial for the management of water resources, irrigation, and water balance and soil conservation. However, it is difficult to effectively simulate its variation due to the complex interactions between land and atmosphere systems as it depends upon many factors.
methods include U.S. Weather Bureau Class-A Pan measurements. The indirect methods, in increasing order of complexity and data requirement, include temperature based formulae (e.g. Blaney-Criddle method), temperature and radiation based formulae (e.g. Priestley-Taylor method), combination formulae which also includes the variation of evaporation with wind velocity and humidity.

A number of researchers have attempted to estimate the evaporation values from climatic variables (Stephens and Stewart, 1963; Linarce, 1967; Burman, 1976; Reis and Dias, 1998; Coulomb et al., 2001; Gavin and Agnew, 2004), and most of these methods require data that are not easily available. Simple methods that are reported (e.g., Stephens and Stewart, 1963) try to fit a linear relationship between the variables. However, the process of evaporation is highly non-linear in nature, as it is evidenced by many of the estimation procedures. Many researchers have emphasized the need for accurate estimates of evaporation in hydrologic modeling studies (Sudheer et al., 2002).

In the last decades, artificial neural networks (ANNs) have been successfully applied in water resources management. Recent experiments have reported that ANN may offer a promising alternative in the hydrological context (Cancelliere et al., 2002; Cigizoglu and Kisi, 2005,2006; Cobaner et al., 2009; Guven and Kisi, 2011; Keskin and Terzi, 2006; Kisi, 2008a,b, 2009a,b, 2010; Kisi and Yildirim, 2005a,b; Kumar et al., 2004; Piri et al., 2009; Sudheer et al., 2002; Supharatid, 2003; Tabari et al., 2010; Tan et al., 2007; Tayfur, 2002). Sudheer et al. (2002) used a feed forward ANN to estimate evaporation and found that the ANN compared favorably to other conventional approach. Keskin and Terzi (2006) developed feed forward ANN models for modeling daily evaporation and found that the ANN model performed better than the conventional method. Tan et al. (2007) used an ANN technique for modeling hourly and daily open-water evaporation rates. Piri et al. (2009) estimated daily evaporations in a hot and dry climate by ANN models. Kisi (2009b) modeled daily pan evaporations using three different neural network techniques, multi-layer perceptron’s (MLPs), radial basis neural networks (RBNNs) and generalized regression neural networks (GRNNs) and he found that the MLP and RBNN could be employed successfully to model the evaporation process using the available climatic data. Tabari et al. (2010) compared ANN and multivariate non-linear regression techniques for modeling daily pan evaporation and found that the ANN performed better than the non-linear regression. Guven and Kisi (2011) modeled daily pan evaporations using linear genetic programming and ANN models. Shirin and Kisi (2011) applied ANN and ANFIS techniques to model daily pan evaporation by using available and estimated climatic data. Nourani et al. (2012) applied three different artificial neural networks (ANNs) viz.: multi-layer perceptron (MLP), radial basis neural network (RBNN) and Elman network for estimating daily evaporation and results denoted the superiority of the ANN models on the classic models.

In this study, an attempt has been made to develop ANN and MLR based evaporation estimation models using climatic parameters as inputs and evaporation as output for Udaipur of Rajasthan, India with the following objectives:
(i) To develop ANN based weekly evaporation estimation model for study area
(ii) To develop MLR based weekly evaporation estimation model for study area
(iii) To evaluate performance and adequacy of developed models

2. Material and Methods
2.1 Description of Study Area and data
Udaipur falls in tropical climatic zone. Geographically it is located at 24.52°N latitude and 73.67°E longitude and an altitude of 598 m above mean sea level. Generally, monsoon starts in the last of June and continues up-to September. The mean annual rainfall is 637 mm of which 80-90 percent occurs during June to September. The mean relative humidity remains almost 80-90 percent from mid-June to February end. The weekly evaporation data for the year 2003 to 2012 (520 weeks) were collected from Meteorological Observatory, Udaipur, India.

2.2 Description of data:-
The weekly evaporation data for the year 2003 to 2012 (520 weeks) were collected from Meteorological Observatory, Udaipur, India. Udaipur falls in tropical climatic zone. Geographically it is located at 24.52°N latitude and 73.67°E longitude and an altitude of 598 m above mean sea level. Generally, monsoon starts in the last of June and continues up-to September. The mean annual rainfall is 637 mm of which 80-90 percent occurs during June to September. The mean relative humidity remains almost 80-90 percent from mid-June to February end.

Fig. 2.1 Udaipur Map

2.3 Development and Implementation of Evaporation estimation models
The weekly values of meteorological data such as temperature (T), wind speed (W), relative humidity (Rh), sunshine hours(S) and evaporation (E) for the year 2003 to 2012 (520 weeks) were collected from Meteorological Observatory, Udaipur, Rajasthan, India. For model development weekly data of temperature (T), wind speed (W), relative humidity (Rh) and sunshine hours(S) were...
taken as inputs to models and current week’s evaporation considered as output of models. Data of years 2003-2009 and 2010-2012 were respectively used for the models development (training of the network) and verification of the models (testing of the network).

2.3.1 Artificial Neural Networks (ANN)

(i) Basic concepts of artificial neural networks

An ANN is an information-processing system composed of many nonlinear and densely inter-connected processing elements or neurons. The main function of the ANN paradigms is to map a set of inputs to a set of outputs. A single processing unit or neuron is shown in Fig.2.2. The incoming signals are multiplied by respective weights through which they are propagated toward the neurons or node, where they are aggregated (summed up) and the net input is passed through the activation function to produce the output.

Fig.2.2: A single artificial neuron (perceptron)

Let $x_i (i = 1, 2, \ldots n)$ are inputs and $w_i (i = 1, 2, \ldots n)$ are respective weights. The net input to the node can be expressed as

$$net = \sum_{i=1}^{n} x_i w_i$$

… (2.1)

The net input is then passed through an activation function $f(.)$ and the output $y$ of the node is computed as

$$y = f(net)$$

… (2.2)

Tanhaxon function is the commonly used nonlinear activation function.

This activation function is shown in Fig.2.3.

Fig.2.3 Tanhaxon function

There are two fundamental classes of network architecture: (i) recurrent neural network and (ii) feed-forward multilayer perceptron (MLP). A recurrent network distinguishes itself from feed-forward network in that it has at least one feedback loop. The basic structure of a feed-forward multilayer network usually consists of three or more layers: the input layer, where the input data is introduced to the network; the hidden layer or layers, where data is processed; and the output layer, where the result of the given input is produced. In these networks, the signal passes in forward direction from input layer to output layer through hidden layer(s). Feed-forward networks are described here in detail as these networks were used in the present work.

(ii) Multilayer feed-forward network

The most important attribute of a multilayer feed-forward network is that it can learn a mapping of any complexity (Zurada. 1992). This network is made up of multiple layers of neurons. In this architecture, besides the input layer and the output layer, the network also has one or more than one intermediate layer(s) called hidden layer(s). Each layer is fully connected to the preceding layer by interconnection strengths or weights. Fig.3.4 illustrates this type of network consisting of a single hidden layer. As can be seen, the generic feed-forward network is characterized by the lack of feedback. Even though this network has no explicit feedback connection when the input is mapped into the output, the output values are often compared with the desired output values, and also an error signal can be employed for adapting the network’s weights during the learning process.

Fig.2.4: Multilayer artificial neural network

(iii) Learning in multilayer feed-forward networks

Learning is a process of forcing a network to yield a particular response to a specific input. In this process, free parameters, i.e. weights of the ANN are adapted by repeated presentations of the training samples. There are two types of learning, supervised and unsupervised. In supervised learning, a set of input pattern and its known output pattern is used to train the network. An external teacher finds the error between computed output and desired output during the training and this error is used to make adjustments in the weights to minimize the error. On the other hand, there is no teacher present to train the patterns in unsupervised learning. Here the system learns of its own detecting regularities in the input space through correlation, without direct feedback from the teacher. Supervised learning is used in the applications involving classification, functional mapping etc.; whereas unsupervised learning is employed in clustering type of applications. As the present work involves functional mapping, it is mandatory to use the supervised learning.
There are several algorithms for supervised learning. Among these algorithms, back-propagation is the most popular due to its simplicity and effectiveness. The back-propagation algorithm is explained below.

**Back-Propagation Algorithm (BP)**

In the back-propagation algorithm, the network weights are modified by minimizing the error between desired (targets) and calculated (predicted) outputs. This algorithm is based on the error-correction learning rule.

**Forward propagation of signals**

**Back propagation of errors**

**Fig.2.5: Directions of signal flow in a multilayer ANN**

Fig.2.6 shows the directions of signal flow and error propagation in a multilayer artificial neural network. Back-propagation is an iterative learning process in which all weight parameters are randomly initialized and then updated (in each iteration) through feed-forward calculations and back-propagation of errors.

**(a) Feed-forward calculation**

In the feed-forward calculation, the nodes in the input layer receive the input signals which are passed to the hidden layer and then to the output layer. The signals are multiplied by the current values of weights, and then the weighted inputs are added to yield the net input to each neuron of the next layer. The net input of a neuron is passed through an activation function to produce the output of the neuron. Considering the ANN shown in Fig.3.5, the procedure for feed-forward calculations in different layers is as follows.

The net input to \( j^{th} \) node of the hidden layer is given by

\[
\text{neth}_j = \sum_{i=1}^{ni} w_{ji} x_i 
\]

where, \( ni \) is the number of neurons in the input layer and \( w_{ji} \) is the connection weight between \( i^{th} \) node of the input layer and \( j^{th} \) node of the hidden layer. The output of \( j^{th} \) node of the hidden layer \( h_j \) is

\[
h_j = f(\text{neth}_j)
\]

where, \( f(.) \) is the activation function, e.g. a tanh-axon activation function.

Similarly, the net input to \( k^{th} \) node of the output layer is given by

\[
\text{net}_k = \sum_{j=1}^{nh} w_{kj} h_j 
\]

where, \( nh \) is the number of neurons in the hidden layer and \( w_{kj} \) is the connection weight between \( j^{th} \) node of the hidden layer and \( k^{th} \) node of the output layer.

The output of \( k^{th} \) node of the output layer is

\[
y_k = f(\text{net}_k)
\]

after calculation of these outputs the error between desired and calculated output is computed which is propagated in the backward direction, as explained below.

**(b) Error back-propagation**

The error calculated at the output layer is propagated back to the hidden layers and then to the input layer, in order to determine the updates for the weights. This method is derived from the well-known gradient descent method in which the weights updation is performed by moving in the direction of negative gradient along the multidimensional surface of the error function. The sum square error \( E \) for a single input-output pair data set is given by

\[
E = \frac{1}{2} \sum_{k=1}^{no} (y_k - t_k)^2 
\]

where, \( t_k \) is the desired output or target at the \( k^{th} \) node and \( y_k \) is the calculated output at the same node.

In order to minimize the above error function, weights are updated by subtracting incremental changes in the weights from their old values. That is,

\[
\Delta w_{kj} = w_{kj} - \Delta w_{kj}^{new} = w_{kj}^{old} - \Delta w_{kj}
\]

\[
w_{kj} = w_{kj}^{old} - \Delta w_{kj}
\]

Where, \( \Delta w_{kj} \) and \( \Delta w_{kj} \) are incremental changes in the weights for output layer and hidden layer respectively.

The learning process starts with a random set of weights. During the training process, weights are updated through error back-propagation.

**(iv) Selection of network architecture**

One of the most important attributes of a layered neural network design is choosing the architecture (Zurada, 1992). The number of input nodes is simply determined by the dimension of the input vector to be generalized or associated with a certain output quality. The dimension of the input vector corresponds to the number of distinct feature of the input pattern. Similarly the number of neurons in the output layer can be made equal to the dimension of vectors to be associated. The size of the hidden layer is the most important consideration when solving the actual problems using multilayer feed-forward neural networks. The most popular and effective strategy for selecting the appropriate number and size(s) of the hidden layer(s) is trial-and-error procedure. A number of networks with one or two hidden layer(s) are trained with different combination of hidden neurons and a network is selected that yields the minimum root mean square error (MSE) and maximum correlation coefficient. It is also important that the size of the network should be as small as possible.

**2.3.2 Multiple Linear Regressions**

In this study, multiple linear regressions were used to estimate the pan evaporation for the study area. Multiple linear regressions (MLR) is a multivariate statistical technique used to model the linear correlations between a single dependent variable \( Y \) and two or more independent variables. The regression equation of \( Y \) can be written as:
\[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_n \quad \ldots (2.10) \]

where, \( Y \) is the response variable; \( X_1, X_2, \ldots, X_n \) are the independent variables; and \( \beta_0, \beta_1, \beta_2, \ldots, \beta_k \) are the regression coefficients.

![Multiple Linear Regression](Fig.2.6)

**3. Results**

The weekly data temperature (T), wind speed (W), relative humidity (RH) and sunshine hours (S) were taken as input to model and evaporation considered as output of models.

Data during years 2003 to 2009 were used for the training (calibration) of the model, whereas the data of years 2010-2012 were used for verification (testing) of the developed models. Qualitative performance of the models have been checked by the graphical comparison, whereas, quantitative performance was verified by estimating the values of statistical indices such as correlation coefficient (CC), root mean squared error (RMSE). The models having high values of correlation coefficient and lower values of RMSE is considered as the better fit model.

**3.1 Artificial Neural Network (ANN) Based Evaporation Estimation Model**

The estimation of evaporation has been carried out using the ANN model for Udaipur location, which falls in tropical climatic zone of India. The ANN model has been developed using the weekly data of temperature (T), wind speed (W), relative humidity (RH) and sunshine hours (S) as a set of inputs and weekly evaporation (E) as output for the network.

Momentum back propagation algorithm was used to train the network for the evaporation estimation with tanh activation function by ANN Network. Various networks of single and two hidden layers were trained for a maximum iteration of 1000, with different combination of hidden neurons and the best suited network was selected based on the minimize values of root mean square error (RMSE), and maximum value of correlation coefficient (CC). The values of statistical indices are given in Table 3.1 for artificial neural network.

**3.2 Performance Evaluation of the Developed ANN Model**

The visual observations and quantitative evaluation of the developed model was performed to judge the goodness of fit between observed and predicted values. The developed model has been verified for its performance considering the runoff and sediment data of monsoon season for years 2010-2012.

**3.2.1 Visual observations evaluation**

The visual observation based on the graphical comparison between the observed and the estimated values is one of the simplest methods for the performance assessment of a model. The performance of the model was evaluated by comparing ordinates of observed and estimated evaporation graphs. The observed and estimated values of evaporation for training period (2003 to 2009) for selected network are depicted in Fig. 3.1, Fig. 3.2, Fig. 3.3, Fig. 3.4, Fig. 3.5, and Fig. 3.6. There is a fairly good agreement between the estimated and the observed evaporation, and overall shape of the plot of estimated evaporation is similar to that of the observed evaporation. Therefore, the visual observations performance during the training has been found satisfactory.

For assessing the suitability of the model, the estimated evaporation was compared visually with the observed evaporation. The observed and the estimated values of evaporation during testing period (2010 to 2012) with selected network are shown in Table 3.1. It is observed from these Tables and figures that there is a close agreement between observed and estimated values of evaporation. Thus, developed model may be regarded as satisfactory on the basis of the visual observations evaluation.

**3.2.2 Quantitative evaluation**

The quantitative evaluation of the model was performed by applying statistical indices, viz., as correlation coefficient (CC), root mean square error (RMSE).

(i) **Root mean square error**

The comparison between the model’s simulated responses to that of recorded watershed response was made by evaluating root mean square error of the models. The root mean square error is zero for perfect fit and increased values indicate higher discrepancies between predicted and observed values.

The root mean square errors of the models were computed by the following equation:

\[ \text{RMSE} = \sqrt{\frac{\sum_{j=1}^{n} (Y_j - Y_{ej})^2}{n}} \quad \ldots (3.1) \]

Where,

- \( Y_j \) = observed values
- \( Y_{ej} \) = predicted values and
- \( n \) = number of observations

(ii) **Correlation coefficient**

The correlation coefficient is an indicator of degree of closeness between observed and predicted values. If observed and predicted values are completely independent, the correlation coefficient will be zero (Mutreja, 1992). The correlation coefficient was computed by the following equation:

\[ \text{CC} = \frac{\sum_{j=1}^{n} \left( Y_j - \bar{Y} \right) \left( Y_{ej} - \bar{Y}_{ej} \right)}{\sqrt{\sum_{j=1}^{n} \left( Y_j - \bar{Y} \right)^2 \sum_{j=1}^{n} \left( Y_{ej} - \bar{Y}_{ej} \right)^2}} \times 100\% \quad \ldots (3.2) \]

where,
\( \bar{Y} \) = mean of observed values
\( Y_{ej} \) = mean of predicted values
\( \bar{Y}_j \) = observed values
\( \bar{Y}_j \) = predicted values and
\( n \) = number of observations

Quantitative performance evaluation of the developed model for training and testing are shown in table 3.1 and table 3.2

<table>
<thead>
<tr>
<th>Networks</th>
<th>Training period</th>
<th>Testing period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>CC</td>
</tr>
<tr>
<td>4-2-1</td>
<td>0.892</td>
<td>0.971</td>
</tr>
<tr>
<td>4-4-1</td>
<td>0.816</td>
<td>0.968</td>
</tr>
<tr>
<td>4-6-6-1</td>
<td>0.836</td>
<td>0.970</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Training period</th>
<th>Testing period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>1.028</td>
<td>0.941</td>
</tr>
</tbody>
</table>

**Fig. 3.1** Observed evaporation and estimated evaporation for training period by ANN for hidden layer 1 and processing element 2.

**Fig. 3.2** Observed evaporation and estimated evaporation for testing period by ANN for hidden layer 1 and processing element 2.

**Fig. 3.3** Observed evaporation and estimated evaporation for training period by ANN for hidden layer 1 and processing elements 4.

**Fig. 3.4** Observed evaporation and estimated evaporation for testing period by ANN for hidden layer 1 and processing elements 4.
3.3 Multiple Linear Regression (MLR) Based Evaporation Estimation Model

Multiple linear regressions were used to estimate the pan evaporation for the study area. Multiple linear regressions (MLR) is a multivariate statistical technique used to model the linear correlations between a single dependent variable \( Y \) and two or more independent variables. The regression equation of \( Y \) can be written as:

\[
y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n
\]

where, \( y \) is the response variable; \( x_1, x_2, \ldots, x_n \) are the independent variables; and \( \beta_0, \beta_1, \beta_2, \ldots, \beta_n \) are the regression coefficients.

3.4 Performance Evaluation of the Developed MLR Model

In this study, the effects of meteorological parameters such as air temperature, relative humidity, wind speed and number of sunshine hours on pan evaporation. The prediction equations decided parameters developed through multiple linear regression analysis. The study reveals the above meteorological parameters can well be correlated with pan evaporation.

The air temperature, wind speed, relative humidity and sunshine hours were taken as the independent variables and evaporation was taken as dependent variable. The multiple linear equation was:

\[
E = 0.1396 + 0.1911 W + 0.2108 S + 0.2816 T - 0.0738 Rh
\]

3.4.1 Qualitative evaluation

The visual observations and quantitative evaluation of the developed model was performed to judge the goodness of fit between observed and predicted values. The visual observation based on the graphical comparison between the observed and the estimated values is one of the simplest methods for the performance assessment of a model.

Conclusions

The present study led to the following conclusions:

1. Meteorological data temperature (T), wind speed (W), relative humidity (Rh), sunshine hours(S) and evaporation (E) were used as inputs for the networks. It is found that artificial neural networks are able to map the input-output relationship for evaporation estimation of Udaipur and showed usefulness of these meteorological variables to map evaporation efficiently.

2. Artificial neural network was able to map the input-output relationship for evaporation estimation of Udaipur. Four meteorological inputs temperature (T), wind speed (W), relative humidity (Rh), sunshine hours(S) were found to be sufficient to provide the hydrological and meteorological history as the inputs to the models.

3. A visual comparison between observed and estimated evaporation graph showed a fair agreement between observed and estimated evaporation data.

4. Based on the performance evaluation indices Networks (4-2-1), (4-4-1), (4-6-6-1), (4-5-5-5-1) were found to be sufficient to provide the hydrological and meteorological history as the set of inputs to the model. Therefore, networks with four number of inputs (temperature (T), wind speed (W), relative humidity (Rh), sunshine hours(S)) were needed for the evaporation estimation in the present study.
References


Evaporation modeling with limited meteorological observations and seasonal effects of meteorological and associated climatic zones in eastern Australia, Evapotranspiration Index using hydro-climatic and climate indices in eastern Australia, Evaporation: energy budget estimates and CRLE model parameters and climate indices in eastern Australia,


Source of support: Nil; Conflict of interest: None declared