MULTILAYER PERCEPTRON-MULTIACTIVATION FUNCTION ARTIFICIAL NEURAL NETWORK MODEL FOR MUNICIPAL WATER DEMAND FORECASTING

Jowhar R. Mohammed[1] and Hekmat M. Ibrahim[2]
[1]Water Resources Engineering, Faculty of Engineering and Applied Science, University of Dohuk, Iraq
[2]Dams and Water Resources Engineering, Faculty of Engineering, University of Sulaimani, Iraq
E-Mail: jowharus@yahoo.com

ABSTRACT

In this research, a multilayer perceptron neural network model with multiactivation function called (MLP-MAF) model has been developed for municipal water demand forecasting. The developed model uses different activation functions in the hidden layer neurons. Different combinations of the linear, logistic, tangent hyperbolic, exponential, sine and cosine activation functions were used in the hidden layer neurons. In order to assess the credibility of developed model results, the model was run over the available data which include the time series of daily and monthly municipal water consumption for fourteen years (1/1/1992 - 31/12/2004) of Tampa city, USA. Each time series was divided into two subsets: the estimate subset for fitting the model and the holdout subset for evaluating the forecasting ability of the model. Additionally, three statistical measurements, namely the coefficient of determination (R^2), the root mean square error (RMSE) and the mean absolute percent error (MAPE) and two hypothesis tests, namely the *t*-test and *F*-test have been reported for examining the forecasting accuracy of the developed model. The results show that the combination of linear, sine and cosine functions is better than other combinations. Furthermore, the effectiveness assessment of this model shows that this approach is considerably more accurate and performs better than the traditional multilayer perceptron (MLP) and radial basis function (RBF) neural networks.

Keywords: forecasting, ANN, water demand.

INTRODUCTION

The scarcity of water is considered the most challenging problem that is facing the countries in the world. The long term forecasting of municipal water demand in such countries is a critical and essential factor for water supply planning, which includes the determining of type, size, location and timing of the required improvements and developments of the water supply systems. On the other hand, short term forecasting of municipal water demand is required for water utilities to proactively optimize pumping and treatment operations to minimize energy, water supply and treatment costs while maintaining a reliable and high quality product for their customers.

There are considerable amounts of published material dealing with water demand forecasting, all of which cannot be reviewed here. However, some principal contributions of historical interest will be cited. In the last few decades there has been a growing scientific interest in the development and use of water consumption forecasting models with monthly, weekly, daily and hourly time scales. Different mathematical models have been investigated and developed by a number of researchers, including regression models that estimate the water consumption as a function of economic and climatic variables and time series models. The autoregressive and autoregressive integrated moving average based models for forecasting of urban water consumption were also developed and compared. Several investigators proposed artificial neural networks as water consumption forecasting models with climatic variables and additional seasonal indices being the model inputs and these models have become prominent for water demand forecast as the neural network was found to outperform the regression and time-series models in some studies. Other models were proposed that explicitly take into account the seasonal, weekly and daily periodicity of urban water consumption. On the other hand, some studies can be found on the application of fuzzy logic, adaptive neuro-fuzzy inference system and wavelet approach for water demand forecasting.

Jain, *et al*. (2001) investigated a relatively new technique of Artificial Neural Networks (ANNs) for using in forecasting short term water demand. The ANN models consistently outperformed the regression and time series models developed in their study and complex ANN performs better than simple ANN. Zhang, *et al*. (2001) presented an experimental evaluation of neural networks for nonlinear time series forecasting. They focused on the feed forward multilayer networks which is known as the multilayer perceptrons (MLPs) and the effects of three important factors (the number of input nodes, the number of hidden nodes and the training sample size) on the neural network forecasting ability were examined by conducting a trail. They found that both the number of input nodes and the number of hidden nodes have significant effects and the number of observations (training sample size) has limited effects on ANN model building and predictive ability. Kim, *et al.* (2001) established an effective model for forecasting the daily urban water demand using feed forward neural networks with genetic algorithms. Alaa and Nisai (2004) presented an approach for short-term (daily) forecasting of municipal water use that utilizes a deterministic smoothing algorithm to predict monthly water use. A two-step approach is employed whereby monthly forecasts are first developed using an adaptive exponential smoothing algorithm, then the daily forecasts is developed by forecasting daily deviations from the monthly average using a time-series regression model. Bougadis, *et al*. (2005) investigated a relative performance of regression, time series analysis and artificial neural network (ANN) models for short-term peak water demand forecasting. In their study, the Fourier analysis for detecting the seasonal and periodic components of time series was used. They found that the ANN technique substantially outperformed regression and time-series methods in terms of accuracy of forecasting. Zhang, *et al*. (2006) developed ANN models for short-term water demand forecasting with a detailed ANN modeling procedure, including identification and selection of input variables and training process. Two ANN models, one for the summer season, and another for the winter season, were developed independently, with different input variables. They found the concept of a division of summer and winter seasons were valid because it simplified the winter season model. Chang and Makkeasorn (2006) proposed and investigated a methodology to predict the water demand with respect to monthly scales in a major metropolitan area using population growth and land use/economic change estimations (i.e., real estate property tax variations) for two types of cities, (fast growth and slow growth) in the Great Orlando Metropolitan Area in Florida. The data (population, water consumption, land use, average incomes, etc.) are projected forward using ANNs models in the first stage analysis and the second stage analysis of their work aims at conducting a forecasting analysis using multilayer perceptron-based neural network models to forecast the future water demand. They concluded that the ANN model would create more accurate predictions if a better database was used. Zhang, *et al*. (2007) employed an ensemble technique to integrate the uncertainty associated with weather variables in short-term water demand forecasting models to provide more reliable and reasonable forecasts. They found that the ensemble forecasting results, compared to the single final forecast computed from the usual deterministic models, improves the robustness of the prediction and the confidence band obtained from the ensemble model provides more reliable. Msiza, *et al*. (2007) compared the efficiency of Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) techniques in water demand forecasting. They found that the ANNs perform significantly better than SVMs. Also Msiza, *et al*. (2007) in another study investigated the multilayer perceptron (MLP) and the radial basis function (RBF) artificial neural networks for forecasting both short-term and long-term water demand in the Gauteng Province, in the Republic of South Africa. They found that the most optimum approximation is the RBF with (r^4logr) activation function. It was observed that the RBF converges to a solution faster than the MLP and it is the most accurate and the most reliable tool in terms of processing large amounts of non-linear, non-parametric data. Adamowski (2008) developed and compared multiple linear regressions, time series analysis, and artificial neural networks (ANNs) as techniques for peak daily summer water demand forecast modeling. The artificial neural network approach was shown to provide a better prediction of peak daily summer water demand than multiple linear regression and time series analysis. Ghiassi, *et al*. (2008) developed a dynamic artificial neural network model (DAN2) for comprehensive (long, medium, and short term) urban water demand forecasting. The DAN2 neural network model employs a different architecture than the traditional Feed Forward Back Propagation (FFBP) model and developed to forecast monthly demand values. Results have shown that DAN2 models outperformed ARIMA and a FFBP-based ANN across all time horizons. Firat, *et al*. (2009) evaluated three artificial neural network techniques, Generalized Regression Neural Networks (GRNN), Feed Forward Neural Networks (FFNN) and Radial Basis Neural Networks (RBNN) based on their performance in forecasting monthly water consumptions from several socio-economic and climatic factors, which affect water use. They found that the model based on multiple input variables performances is better than the model based on a single input variable and Generalized Regression Neural Networks (GRNN) method is better than other ANNs and Multiple Linear Regression (MLR) models and can be successfully applied to establish accurate and reliable water consumption forecasting models. Leon F. and Zaharia M. H. (2010) proposed a stacked hybrid neural network model for time series forecasting. The model contains one normal multilayer perceptron with bipolar sigmoid activation functions and the other with an exponential activation function in the output layer. They found that the stacked hybrid neural network model performs well on a variety of benchmark problems for time series.

The overall objective of this research is to develop an accurate model to forecast short and long term municipal water demand and apply it to the available data of water consumption. In order to meet this goal, a multilayer perceptron multi activation function neural network (MLP-MAF) model has been developed. The model differs from the traditional multilayer perceptron neural network that it uses neurons in the hidden layer with different types of activation functions. The model applied to the time series of daily and monthly water consumption of Tampa city, USA.

The study area and available data

Because the development of the water forecasting model is dependent on the availability of water consumption data, therefore, to illustrate the applicability and capability of the model developed in the present research is applied to area, which can access to the required data. The area that the required data have been obtained for and the developed model applied to it as a case study is Tampa in the USA. The available data includes the daily water consumption and climatological data (minimum and maximum temperature, rainfall, mean relative humidity and mean wind speed) for a period of thirteen years from 1-January-1992 to 31-December-2004 and the monthly data were derived from the daily data.

The Tampa Bay Area is the region of west central Florida adjacent to Tampa Bay, USA. Tampa Bay Water is a regional wholesale drinking water utility that serves customers in the Tampa Bay, Florida region, USA as shown in Figure-1. The agency is special districts of the state created by inter local agreement among six member governments. Customers served in the area are predominantly residential users, with commercial, industrial and public consumption included (Asefa and Adams, 2007).

Model performance and accuracy measurements

The most important criterion for evaluating forecasting models or choosing between competing models is accuracy. Generally speaking, the closer the forecasts $\hat{y}$ to the observed values y, of the series, the more accurate the forecasting model is. Thus the quality of a model can be evaluated by examining the series of forecast errors $(y - \hat{y})$. The most commonly used measures of forecast accuracy are mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) (Rumantir, 1995). In addition to these measures, the most commonly used error measures in water resources modeling include the mean squared relative error (MSRE), the coefficient of determination (R^2) and the coefficient of efficiency (CE) (Kingston, 2006).

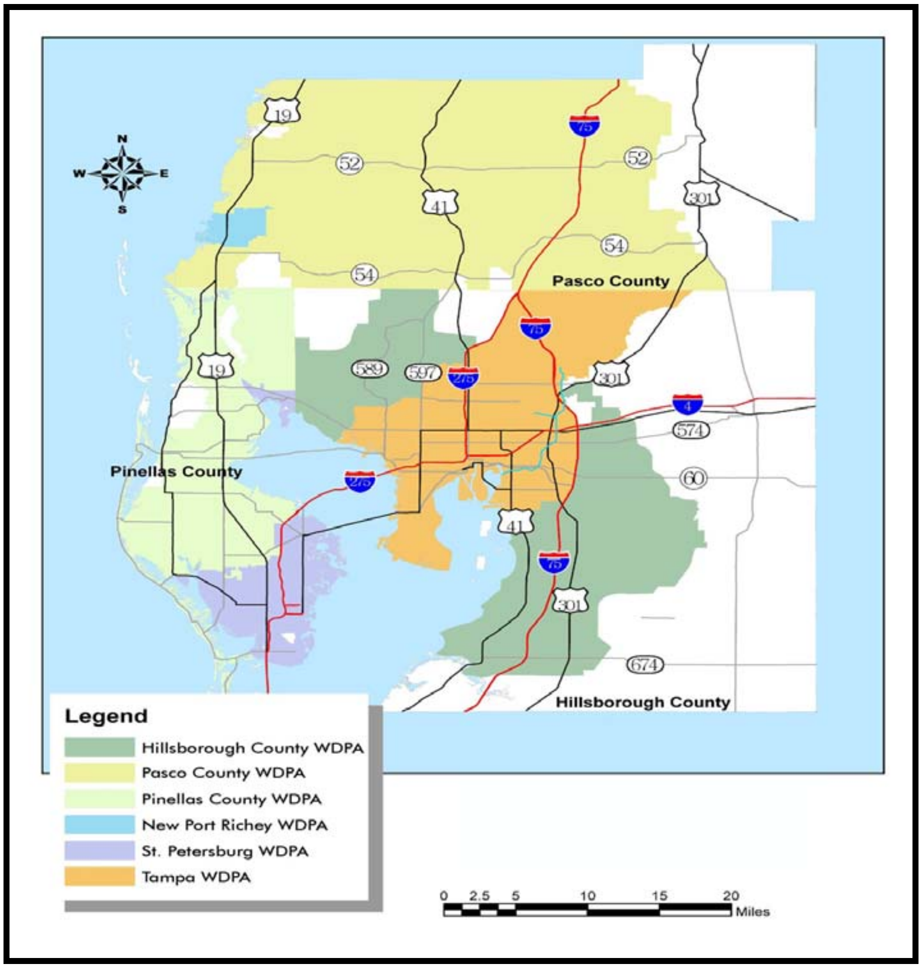

Figure-1. Tampa Bay area map which is used as a study area in the research (Asefa and Adams, 2007).

A more realistic way of assessing a model's accuracy is to use a holdout set, that is, some of the data at the end of the series are omitted before the models are estimated. Then the models are compared on the basis of how well they forecast the data, which have been withheld rather than how well they forecast the same data which has been used for modeling (Makridakis, *et al*., 1998). Therefore, in the present research for the purpose of selecting and comparison of forecasting models, the time series data were split into two sets. The first set of the data is used to estimate the parameters of the particular model (estimate set). Then with these estimates, the model is used to forecast ahead the remaining data points (holdout set). To obtain information about the model's ahead forecasting performance; the resulting out of sample forecasts is compared to the actual holdout series values. Some of the above measures are used for holdout data and the model that yields best values for these statistics on holdout set would be chosen as a good model. For monthly water consumption data, the last 24 months of the data were to be held out, and then the model were fitted on the remaining data and used to forecast 24 months ahead. On the other hand, for daily water consumption data the last 5, 10 and 15 days of the data were held out. According to this partitioning of data, the number of estimation and holdout sets for the data considered in this research will be as shown in Table-1.

Table-1. Duration and estimation with holdout water consumption data sets for Tampa area.

Data	Years	Duration	Estimation data No.	Holdout data No.
Monthly data	1992 - 2004	156 months	132	24
Daily data	1992 - 2004	4745 days	4740, 4735, 4730	5, 10, 15

In order to provide an indication for the performance of the developed models in the present research and goodness of fit between the observed and each of predicted values for estimating set (fit) and holdout set (forecast), R^2, RMSE and MAPE were used for the models investigated in this research as shown in the following equations:

$$R^2 = \left(\frac{\sum_{i=1}^{N} (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2 (\hat{y}_i - \bar{\hat{y}})^2}} \right)^2 \quad \ldots\ldots\ldots \tag{1}$$

$$RMSE = \left(\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \right)^{1/2} \quad \ldots\ldots\ldots \tag{2}$$

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \cdot (100\%) \quad \ldots\ldots\ldots \tag{3}$$

In which N is the number of data points, y_i, $\bar{y}$ are the observed data with its mean, respectively and $\hat{y}$, $\bar{\hat{y}}$ are the corresponding predicted data with its mean, respectively. Also two additional statistical tests, t-test and F-test were used in this research to compare the mean and variance of observed and predicted series for estimating and holdout sets. Among the most frequently used t-tests are a two sample location test of the null hypothesis that the means of two normally distributed populations are equal. The t-test statistic is:

$$t - statistic = \frac{\bar{X}_1 - \bar{X}_2}{S_d \cdot \sqrt{\frac{N_1 + N_2}{N_1 . N_2}}} \quad \ldots\ldots\ldots\ldots \tag{4}$$

$$S_d = \sqrt{\frac{\sum_{i=1}^{N_1} (X_i - \bar{X}_1)^2 + \sum_{j=1}^{N_2} (X_j - \bar{X}_2)^2}{N_1 + N_2 - 2}} \quad \ldots\ldots\ldots \tag{5}$$

Where N_1 and N_2 are the subseries sizes, X_i is the sample values in the N_1 series and X_j in the N_2 series. The variable t of Equation (4) follows the Student t-distribution with $(N_1 + N_2 - 2)$ degrees of freedom. The critical value t_c for the 95% significance probability level is taken from the Student t-tables. If $|t| > t_c$, then there is a difference in the mean of two series.

In statistics, an F-test for the null hypothesis that two normal populations have the same variance is sometimes used. The expected values for two populations can be different, and the hypothesis to be tested is that the variances are equal. The F-test statistic is:

$$F = \frac{S_y^2}{S_{\hat{y}}^2} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{6}$$

Where S_y^2, $S_{\hat{y}}^2$ are the variance of observed and predicted data, respectively. The observed and predicted sets have an F-distribution with N-2 degrees of freedom if the null hypothesis of equality of variances is true. The null hypothesis is rejected if F is either too large or too small.

Multilayer perceptron neural network (MLP)

The basic architecture of artificial neural network consists of three types of neuron layers: input layer, hidden layers and an output layer. Artificial neurons in one layer are connected, fully or partially, to the artificial neurons in the next layer. Feedback connections to previous layers are also possible (Engelbrecht, 2007).

A multilayer perceptron is feed forward neural network architecture with unidirectional full connections between successive layers. Figure-2 shows the structure of a multilayer perceptron artificial neural network which has an input layer, a hidden layer, and an output layer of neurons. These three layers are linked by connections whose strength is called weight. Thus there are two sets of weights: the input-hidden layer weights and the hidden-output layer weights. These weights provide the network with tremendous flexibility to freely adapt to the data; they are the free parameters, and their number is equivalent to the degrees of freedom of a network (Samarasinghe, 2006).

The output of typical MLP neural networks with a single layer of hidden neurons, as shown in Figure-2, is given by:

$$\hat{y}_k = f_o\left[\sum_{j=1}^{m}\left(w_{k,j} f_h\left(\sum_{i=1}^{n}(w_{j,i}\, x_i) + b_j\right)\right) + b_k\right] \quad \ldots\ldots \tag{7}$$

Where x is the input variable, $\hat{y}_k$ is the output, n is the number of neurons in input layer which is equal to the number of input variables, m is the number of neurons in hidden layer, w is the weights of input-hidden and hidden-output layers, b is the bias of hidden and output layers, f_h is the activation function of hidden layer and f_o is the activation function of output layer (Dreyfus, 2005).

Training of multilayer perceptron artificial neural network basically involves feeding training samples as input vectors through a neural network, calculating the error of the output layer, and then adjusting the weights of the network to minimize the error. The average of all the squared errors for the outputs is computed to make the derivative easier. Once the error is computed, the weights can be updated one by one (Engelbrecht, 2007). The sum of squares error is simply given by the sum of differences between the target and prediction outputs defined over the entire training set. Thus:

$$SSE = \sum_{i=1}^{n}(output_i - target_i)^2 \quad \ldots\ldots\ldots\ldots\ldots \tag{8}$$

Where n is the number of training cases. It is clear that the bigger the difference between predictions of the network and the targets, the higher the error value, which means more weight adjustment, is needed by the training algorithm (Hill and Lewicki, 2007).

There are three main types of learning: supervised, unsupervised and reinforcement learning. The primary interests are the supervised learning algorithms, the most frequently used in real applications, such as the back propagation training algorithm, also known as the generalized delta rule. Two types of supervised learning algorithms exist based on when weights are updated:

- Stochastic/online learning, where weights are adjusted after each pattern presentation. In this case the next input pattern is selected randomly from the training set, to prevent any bias that may occur due to the order in which patterns occur in the training set.
- Batch/offline learning, where weight changes are accumulated and used to adjust weights only after all training patterns have been presented.

Learning iterations which are referred to as epochs, consists of two phases:

a) Feed forward pass, which simply calculates the output value(s) of the neural network for each training pattern.
b) Backward propagation, which propagates an error signal back from the output layer toward the input layer. Weights are adjusted as functions of the back propagated error signal (Engelbrecht, 2007).

It has been proven that back propagation learning with sufficient hidden layers can approximate any nonlinear function to arbitrary accuracy. This makes back propagation learning neural network a good candidate for signal prediction and system modeling (Abraham, 2005).

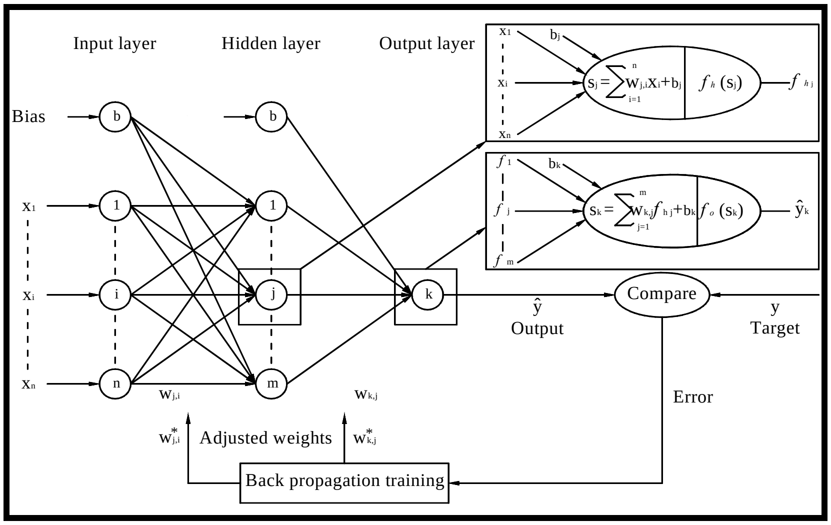

Figure-2. Structure of typical multilayer perceptron artificial neural network.

Radial basis function neural networks (RBF)

A radial basis function neural network is a feed forward neural network where hidden units do not implement an activation function, but represents a radial basis function (Engelbrecht, 2007). The common architecture of a radial basis function neural network firmly consists of three layers: the input layer, the hidden layer, and the output layer. In the neurons of the hidden layer, the activation functions are placed. The input layer of the network does no processing and it is directly connected with the hidden layer of the network, so that only the connections between the hidden layer and the output layer are weighted (Palit and Popovic, 2005). Therefore, the radial basis function neural network can be considered as a two-layer network in which the hidden layer performs a fixed nonlinear transformation with no adjustable parameters, so that the input space is mapped into a new space. The hidden layer in a radial basis function neural network consists of an array of nodes and contains a parameter vector called a center. A distance measure, to determine how an input vector is far from the center, is typically the standard Euclidean distance measure. The node calculates the Euclidean distance between the center and the input vector of the network and the result is passed through a nonlinear function. A transfer function, which is a function of a single variable, determines the output of the distance function. The output layer then combines outputs in the latter space linearly (Taylor, 1996).

Radial basis function neural networks may require more neurons than standard feed forward back propagation networks, but often they can be designed with less time (Abraham, 2005). Figure-3 illustrates a general architecture of the radial basis function neural network. The architecture is very similar to that of a standard multilayer perceptron neural network, with the following differences:

- Hidden units implement a radial basis function.
- Weights from the input neuron to a hidden neuron, referred to as μ which represents the center of the radial basis function.
- Some radial basis functions are characterized by a width *σ*. For such basis functions, the weight from the basis neuron in the input layer to each hidden neuron represents the width of the basis function. Note that input neuron *b* refers to an input signal of +1 (bias).

Each hidden unit implements a radial basis function. These functions, also referred to as kernel functions, are strictly positive, radially symmetric functions. A radial basis function has a unique maximum at its center *μ*, and the function usually drops off to zero rapidly further away from the center. The output of a hidden unit indicates the closeness of the input vector *x*, to the center of the basis function.

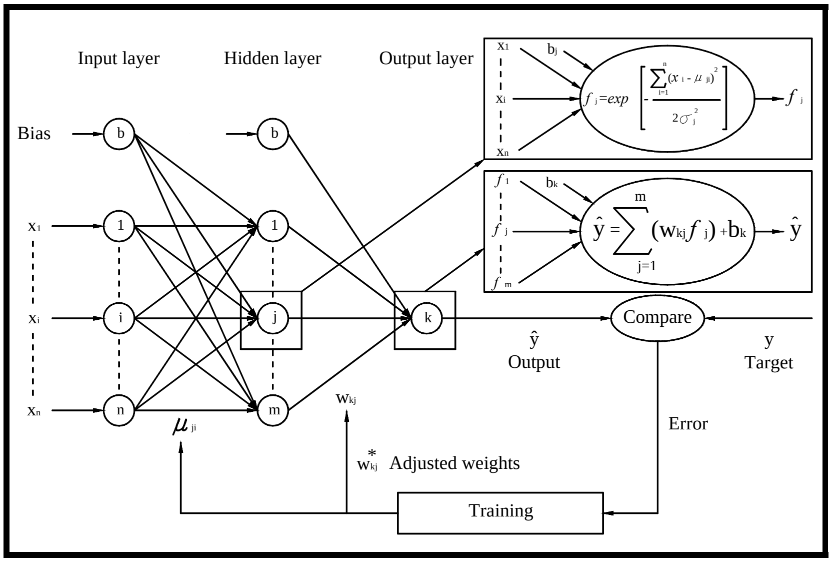

Figure-3. Structure of typical radial basis function artificial neural network.

A number of radial basis functions have been proposed: linear, cubic, thin plate spline, multi-quadratic, Inverse multi-quadratic, Gaussian and logistic functions (Engelbrecht, 2007). The input data are presented the network in input layer and these data are transferred to hidden layer by radial basis function. The response of the network is obtained in the output layer. The various types of radial basis functions can be used for transferring of the input data to the hidden layer (Firat and Gungor, 2007). However, in this research, the Gaussian activation function, the most commonly used, is selected as the activation function for training data set. The mathematical structure of Gaussian activation function is demonstrated as follows:

$$f_j = exp\left(-\frac{\sum_{i=1}^{n}(x_i - \mu_{ji})^2}{2\sigma_j^2}\right) \quad \ldots\ldots\ldots\ldots\ldots \tag{9}$$

Where *x* is the input, *n* is the number of inputs, *μ* is the parameter which is the position of the center of the Gaussian and σ^2 is its standard deviation. The f_j has an appreciable value only in the neighborhood of the center (Taylor, 1996). The standard deviation and center determine the properties of each function during the training process. The response of each hidden neuron is scaled by its connecting weights to the output neurons and then summed to produce the overall network output. Therefore, the output $\hat{y}_k$ of the neural network for Gaussian radial basis functions is given by:

$$\hat{y}_k = \sum_{j=1}^{m} w_{kj} f_j + b_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{10}$$

Where *w* is the weight of connection between the hidden neuron *j* and the output neuron *k*, *b* is the bias and *m* is the number of neurons in the hidden layer (Dreyfus, 2005).

The methods used to train radial basis function neural networks are fundamentally different from those employed for multilayer perceptron neural networks (Hill and Lewicki, 2007). The most important issue here for each neuron in the hidden layers is the selection of the center *u*, the spread around the center *σ* and the weights between the hidden and output neurons. Training of a radial basis function neural network should therefore consider methods to find the best values for the parameters *μ*, *σ* and *w*. The network training process mainly includes two training phases:

- Initialization of radial basis function centers *u*, for instance using unsupervised clustering methods, linear vector quantization, or decision trees.
- Output weight training of the radial basis function network using an adaptive algorithm to estimate its appropriate values.

In some cases, it is recommended to add a third training phase, in which the entire network architecture is adjusted using an optimization method (Palit and Popovic, 2005).

Artificial neural networks generalization and performance

The performance of neural networks is measured by how well they can predict unseen data (an unseen data set is one that was not used during training). This is known as generalization. There is a relation between over fitting the training data and poor generalization (Hill and Lewicki, 2007). That is, neural networks that over fit cannot predict correct output for data patterns not seen during training. Therefore, generalization is a very important aspect of neural network learning (Engelbrecht, 2007).

To avoid over fitting, the flexibility of a neural network must be reduced. Flexibility comes from the hidden neurons and as the number of hidden neurons increases, the number of network parameters (weights) increases as well. On the other hand, there must be enough neurons to avoid bias or under fitting (Samarasinghe, 2006). There are several techniques to combat the problem of over fitting and tackling the generalization issue. The most popular is cross validation which is a traditional statistical procedure for random partitioning of collected data into a training set and a test set (Palit and Popovic, 2005). Test data is a holdout sample that will never be used in training. Instead, it will be used to halt training to mitigate over fitting. The process of halting neural network training to prevent over fitting and improving the generalization ability is known as early stopping.

Sometimes the test data alone may not be sufficient proof of a good generalization ability of a trained neural network. For example, a good performance on the test sample may actually be just a coincidence. To make sure that this is not the case, another set of data known as the validation sample is often used. Just like the test sample, a validation sample is never used for training the neural network. Instead, it is used at the end of training as an extra check on the performance of the model. If the performance of the network was found to be consistently good on both the test and validation samples, then it is reasonable to assume that the network generalizes well on unseen data.

There are a number of main steps in the artificial neural network development process as shown in Figure-4 and there are also a number of options available at each step and, while this provides great flexibility in artificial neural network modeling, it also leaves the modeler faced with the difficult task of selecting the most suitable methods. One of the most important steps in using a neural network to solve real world problems is to collect and transform data into a form acceptable to the neural network. In practice, the simplest and linear transformation are most frequently used and obtained by rescaling or by standardization as follows:

$$x_{ni} = \frac{x_i}{x_{max}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{11}$$

$$x_{ni} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{12}$$

$$x_{ni} = \frac{x_i - x_{mean}}{x_{sd}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \tag{13}$$

Where x_i is the untransformed data, x_{ni} is the transformed data, x_{min}, x_{max} are the minimum and maximum of untransformed data and x_{mean}, x_{sd} are the mean and standard deviation of untransformed data (Palit and Popovic, 2005).

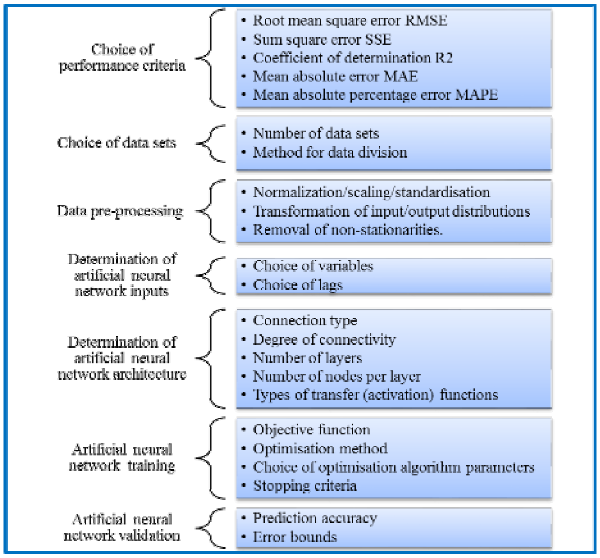
Choice of performance criteria
• Root mean square error RMSE
• Sum square error SSE
• Coefficient of determination R2
• Mean absolute error MAE
• Mean absolute percentage error MAPE
Choice of data sets
• Number of data sets
• Method for data division
Data pre-processing
• Normalization/scaling/standardisation
• Transformation of input/output distributions
• Removal of non-stationarities.
Determination of artificial neural network inputs
• Choice of variables
• Choice of lags
Determination of artificial neural network architecture
• Connection type
• Degree of connectivity
• Number of layers
• Number of nodes per layer
• Types of transfer (activation) functions
Artificial neural network training
• Objective function
• Optimisation method
• Choice of optimisation algorithm parameters
• Stopping criteria
Artificial neural network validation
• Prediction accuracy
• Error bounds

Figure-4. Main steps in the development of an artificial neural network (Kingston, 2006).

For an explanatory or causal forecasting problem, the inputs to an ANN are usually the independent or predictor variables. The functional relationship estimated by the ANN can be written as:

$$y = f(x_1, x_2, \ldots, x_s) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(14)}$$

Where $x_1, x_2, \ldots, x_s$ are *s* independent variables and y is a dependent variable. In this sense, the neural network is functionally equivalent to a nonlinear regression model. On the other hand, for an extrapolative or time series forecasting problem, the inputs are typically the past observations of the data series and the output is the future value. The ANN performs the following function mapping:

$$y_{t+1} = f(y_t, y_{t-1}, \ldots, y_{t-s}) \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(15)}$$

Where y is the observation at time *t*. Thus the ANN is equivalent to the nonlinear autoregressive model for time series forecasting problems. It is also easy to incorporate both predictor variables and time lagged observations into one ANN model, which amounts to the general transfer function model (Zhang, *et al*., 1998).

The developed MLP-MAF ANN model

Generally, a neural network may have different activation functions for different nodes in the same or different layers. Yet, almost all the networks use the same activation function, particularly for the nodes in the same layer. Therefore, this research deals with a developed multilayer perceptron neural network with different activation functions for different nodes in the hidden layer named multilayer perceptron multiactivation function neural network (MLP-MAF). The neural network developed here consists of three layers: input, hidden and output layer as shown in Figure-5. Equal numbers of nodes in the hidden layer have been used for activation functions. A several combinations of the linear, logistic, tangent hyperbolic, exponential, sine and cosine functions in the hidden nodes were investigated. On the other hand, only the linear function was used in the output node.

In order to form and train the model, a code has been written using the MATLAB version 7.0 package software named MLP-MAF code. The training of the neural networks is performed with the traditional backward propagation algorithm. The lagged observations of water consumption were used as inputs to the input layer nodes. The results of the developed model were compared with the results of different traditional MLP and RBF neural networks developed using STATISTICA version 8.0 package software. The neurons of the hidden layer are equally divided into g groups with a different activation function for each one.

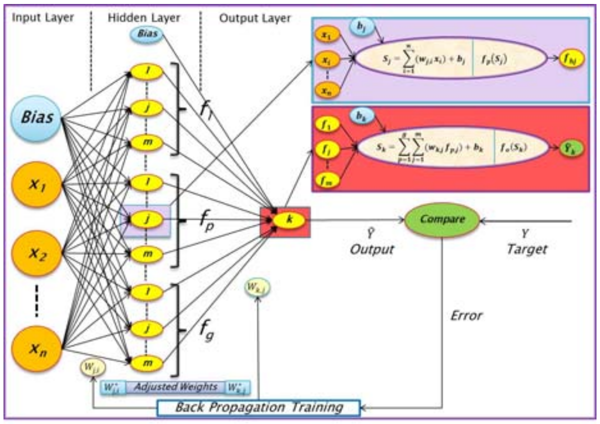

Figure-5. Architecture of multilayer perceptron-multiactivation function (MLP-MAF) artificial neural network.Building of MLP-MAF model.

Basically, the MLP-MAF model predicts the value of the time series y_t at the time step t depending on the past values. The output of the neuron k in the output layer of the network can be expressed as follows:

$$\hat{y}_k = f_o\left[\sum_{p=1}^{g}\left(\sum_{j=1}^{m}\left(w_{jk}f_{pj}\left(\sum_{i=1}^{n}(w_{j,i}\,x_i) + b_j\right)\right) + b_o\right)\right] + \varepsilon_t \quad \ldots \text{(16)}$$

Where *x* is the input variable, $\hat{y}_k$ is the output, *n* is the number of neurons in input layer which is equal to the number of input variables, *m* is the number of neurons in each group in the hidden layer, *g* is the number of groups, w is the weights of input-hidden and hidden-output layers, b is the bias of hidden and output layers, f_p is the activation function of group p in the hidden layer and f_o is the activation function of output layer.

Following the steps of artificial neural network building, described earlier, the methods adopted here for carrying out each step of the MLP-MAF development processes are summarized below:

MLP-MAF network architecture: the input variables in the present study's case are the past values of water consumption time series used to determine the underlying pattern in the time series and to make forecasts for future values. On the other hand, the output layer includes one node corresponding to the present water consumption value in the output vector. Selection of the input variables is one of the most important steps in developing a satisfactory forecasting model, because these variables determine the structure of the forecasting model and affect the weighted coefficient and the results of the model. For forecasting urban water consumption, a number of time lagged values were selected as input variables based on the lag correlations between the present value and lagged values in order to determine the best input structure. The lag correlations between the present value of water consumption and the lagged values of Tampa water consumption time series are shown in Table-2. The time lags that have been used as inputs to the developed MLP-MAF model for Tampa water consumption data are shown in the mentioned table in bold style.

Table-2. Lag correlations between the present water consumption value and time lagged values for Tampa data.

Lag	**Tampa monthly water consumption** (m^3)	**Tampa daily water consumption** (m^3)	**Lag**	**Tampa monthly water consumption** (m^3)	**Tampa daily water consumption** (m^3)
0	1.00	1.00	7	0.01	0.77
1	0.57	0.82	8	-0.09	-
2	0.26	0.68	9	-0.03	-
3	-0.10	0.71	10	0.20	-
4	-0.21	0.69	11	0.34	-
5	-0.08	0.60	12	0.52	-
6	-0.08	0.67	365	-	0.34

In the developed model, only one hidden layer was used, and the trial-and-error procedure was used to select the optimum number of hidden neurons in the hidden layer based on the best generalizability. For each group in the hidden layer, a number of hidden neurons from one to two-time input variables (2N) have been investigated in order to find the optimal neuron numbers.

Activation function: the activation functions, which have been used in the hidden layer of the MLP-MAF developed model, include several combinations of functions. For daily and monthly water consumption time series of Tampa, four combinations of the activation functions were used in the hidden layer. These combinations are as below:

a) Linear, logistic, tangent hyperbolic, exponential and sine (five groups of neurons).
b) Logistic, tangent hyperbolic, exponential and sine (four groups of neurons).
c) Tangent hyperbolic, exponential and sine (three groups of neurons).
d) Linear, sine and cosine (three groups of neurons).

In all the above models, the linear activation function was used in the output neuron of the output layer. The results of the developed models were compared with the results of different traditional MLP and RBF neural networks produced using STATISTICA version 8.0 package software.

Training algorithm: the training of the developed MLP-MAF network is performed with the traditional backward propagation algorithm. The following equation describes the weights adjusting of input-hidden and hidden-output of multilayer perceptron-multiactivation function neural network.

The inputs to the hidden layer neurons are:

$$HIN = ([I]\cdot[WI])^T \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (17)$$

$$HIN = \left(\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \end{bmatrix} \cdot \left[\begin{array}{ccccccc} & Group-1 & & \cdots & & Group-g & \\ WI_{11} & \cdots & WI_{1m} & \cdots & WI_{11} & \cdots & WI_{1m} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ WI_{n1} & \cdots & WI_{nm} & \cdots & WI_{n1} & \cdots & WI_{nm} \end{array}\right]\right)^T . \quad (18)$$

The output of the hidden neurons:

$$HID = \left[f_{h1}\begin{pmatrix} Group-1 \\ [I]^T \times [WI] \end{pmatrix} \ldots\ldots\ldots\ldots f_{hg}\begin{pmatrix} Group-g \\ [I]^T \times [WI] \end{pmatrix}\right]^T \ldots \quad (19)$$

The inputs to the output layer neurons are:

$$OIN = [HID]\cdot[WO] \;\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (20)$$

The output of the output layer neurons is:

$$O = f_O([HID]^T \times [WO]) \text{ ……………………… } \quad (21)$$

Where O is the output vector, I is the input vector, WI is the input-hidden weights, WO is the hidden-output weights and $f_{h1} \cdots f_{hg}$, f_O are the activation functions of hidden and output layer neurons respectively. The weight updates of hidden-output become:

$$WO_{(new)} = WO_{(old)} - \{\eta_a \cdot [HID] \times [Error] \cdot \acute{f}_O([OIN])\} \text{ … } \quad (22)$$

Note that in case of the present research, by using linear function for f_O, the term $\acute{f}_O([OIN])$ will be equal to 1 and the $Error$ is the difference between the outputs and targets.
The weight updates of input-hidden become:

$$WI_{(new)} = WI_{(old)} - \{\eta_b \cdot \{[WO] \times [Error] \cdot \acute{f}_h([HIN]) \cdot [I]^T\}^T\} \quad (23)$$

The term $\acute{f}_h([HIN])$ is calculated as below:

$$\acute{f}_h([HIN]) = \left[\acute{f}_{h1}\begin{pmatrix} Group-1 \\ [I]\cdot[WI] \end{pmatrix} \text{………………} \acute{f}_{hg}\begin{pmatrix} Group-g \\ [I]\cdot[WI] \end{pmatrix}\right]^T \quad (24)$$

In which η_a and η_b are the learning rates for input-hidden and hidden-output weights respectively. Initializing the weights was made randomly in the range (-0.1, +0.1) and choosing the learning rate will be done by trial and error.

Data preprocessing: data preprocessing for the developed MLP-MAF model was achieved by normalization using Equation (12). The time series of water consumption were transformed into the range (-1, +1), before the training process began by the model. On the other hand, the traditional models developed using STATISTICA version 8.0 package software for comparison were used a transformed data in the range (0, 1) which is applied by the software automatically before the training process.

Training and test sets: as mentioned earlier, the entire available data set was divided into estimate and holdout subsets. The estimate subset was divided into training and test samples randomly to be used in MLP-MAF, classic MLP and RBF modeling. The estimating sets of data were divided with 80% of the data allocated to training subset and 20% allocated to the testing subset.

Performance criteria: the performance of the developed MLP-MAF models for estimate and holdout subsets was evaluated according to statistical criteria described earlier. For stopping the training process of the model for estimate subset, three statistical criteria will be entered to the MATLAB code. These statistical criteria are: coefficient of determination (R^2), root mean square error (RMSE) and mean absolute percentage error (MAPE). The stopping of the training process was based on the performance of testing sample of the estimate subset.

MLP-MAF MATLAB code: Because the capability of all available software packages for building neural networks are limited to build neural networks use the same activation function in the same layer, therefore, a MATLAB code has been written using MATLAB version 7.0 package software to build and train the developed multilayer perceptron multiactivation function neural network called MLP-MAF code. Figure-6 illustrates the back propagation algorithm and the main training steps in MLP-MAF MATLAB code.

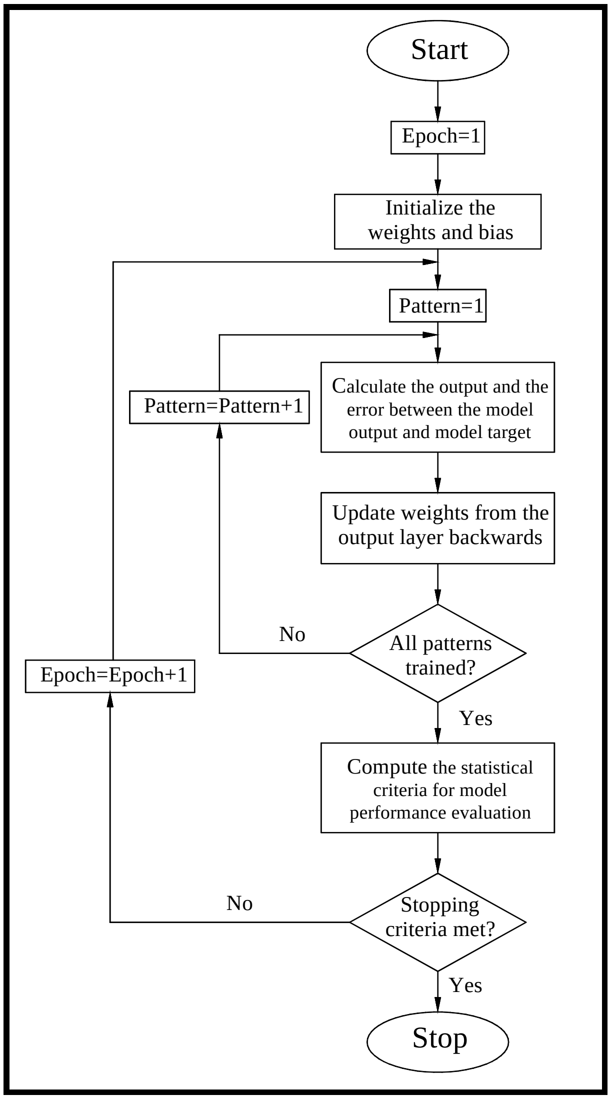
Start
Epoch=1
Initialize the weights and bias
Pattern=1
Calculate the output and the error between the model output and model target
Pattern=Pattern+1
Update weights from the output layer backwards
No
All patterns trained?
Epoch=Epoch+1
Yes
Compute the statistical criteria for model performance evaluation
No
Stopping criteria met?
Yes
Stop

Figure-6. The schema of the back propagation algorithm, outlining the main training steps in MLP-MAF MATLAB code.

MLP-MAF model application

To demonstrate the effectiveness of the proposed MLP-MAF neural network model, it has been applied to two classes of forecasting problems. The first class includes long term forecasting of monthly water consumption time series of Tampa area. The second class of the problem is a short term forecasting of daily water consumption time series of Tampa area. For each class of the problems, the performance of MLP-MAF models have been measured and compared using three criteria: the coefficient of determination (R^2), the root mean squared error (RMSE) and the mean absolute percentage error (MAPE). The MLP-MAF results were compared against the best traditional MLP and RBF models. The same exact set of input data is used in all comparisons and the same percentage of data partitioning was used for all models.

Monthly water consumption time series

As mentioned earlier, the developed MLP-MAF neural network model has been applied to monthly water consumption time series. The input variables to the model are the past values of the time series. To investigate the dependency between past values and present water demand and to determine the lagged values to be used as inputs to the model, lag correlation coefficients between the present monthly demand and past monthly values were calculated and are shown in Table-2. This information was used to aid in selecting input variables for MLP-MAF and traditional MLP and RBF neural networks. This Table shows the strong correlations of present monthly water demand with the lagged values. The lagged values used as inputs to the models are: lags 1, 2, 10, 11 and 12.

Four MLP-MAF models with different combinations of activation functions in the hidden layer, as described earlier, were developed for monthly water consumption time series. On the other hand, the STATISTICA version 8.0 package software was used to develop the traditional MLP and RBF neural network models. For comparison purpose with MLP-MAF models twenty five separate traditional MLP models and one RBF model were developed. Different activation functions were used in the hidden and output layer of traditional MLP models, and Gaussian function was used in the hidden layer of RBF model with linear function in the output layer. All developed ANNs were consisting of an input layer, one single hidden layer and one output layer consisting of one node denoting the predicted water demand.

All the MLP-MAF and traditional MLP and RBF models used the same input variables (lagged values) and first, trained using the data in the estimate subset to obtain the optimized set of connection strengths and then evaluated using the data in the holdout subset and compared using the statistical measures of goodness of fit that were described earlier. The architecture of the developed models and the performance measures of the results for monthly water consumption time series are shown in Table-3. As seen in this table, the models are evaluated based on their performance in estimate and holdout sets. The models have shown variations in the performance evaluation. Comparing the results of MLP models with RBF model, generally, the performance of MLP models is better than RBF model. The correlations of determination values (R^2) for MLP models are higher than that for RBF model and MAPE with RMSE values for MLP models are lower than that for RBF model for both estimate and holdout sets.

Comparing the results of MLP models for the estimate set, it is seen that the performance of the models with tangent hyperbolic activation function in the hidden layer is better than other models, while for the holdout set, the results of the models with logistic activation function in the hidden layer, in general, are better than other MLP models. On the other hand, generally, the performance of MLP-MAF models is better than other models for the estimate set. In addition, for the holdout set, the MLP-MAF models performances are better than the traditional ANN models. The performance of model M30, which consists of three activation functions (linear, sine, cosine) in the hidden layer, is better than other MLP-MAF models for the holdout set of data. Furthermore, it is seen that the performance of M27 model is close to M28 model. In addition to R^2, MAPE and RMSE measurements, the *t* value was also used to test the hypothesis that MLP-MAF and traditional ANN models forecasts have the same means of the observed values. Comparing the *t*-test values for MLP-MAF models with critical *t* values indicate that the hypotheses are not statistically significant. The results of the models demonstrate that MLP-MAF can be applied to forecast urban water consumption. Comparisons of results of four MLP-MAF models for monthly water consumption with the observation data records are shown in Figure-7.

An overall conclusion of MLP-MAF and traditional MLP and RBF applied to Tampa monthly water consumption data indicates that none of the methods was performed very well in forecasting monthly urban water demand. This seems to indicate that there were problems with the data used or the wrong deriving input variables were used. The present research focused on the modelling of monthly water demand forecasts using past water consumptions as inputs to the developed models. The models could potentially be improved if other variables such as socioeconomic and climatic that affect water demand were used. Therefore, a further study recommended to be carried out in the future on the same and other data using socioeconomic and climatic variables in addition to past water demands as input variables in the MLP-MAF models.

Table-3. Structure and statistical performance measurements of MLP-MAF, traditional MLP and RBF models of monthly water consumption time series for Tampa.

Model	Hidden layer activation function/s	Output layer activation function/s	**Estimate set** (Observed - Fit)					**Holdout set** (Observed - Forecast)				
			R^2	MAPE (%)	RMSE	*t*-value	*F*-value	R^2	MAPE (%)	RMSE	*t*-value	*F*-value
M1: MLP (5-5-1)	Linear	Linear	0.422	6.993	1789807	0.400	2.995	0.353	4.728	1078685	0.586	3.623
M2: MLP (5-10-1)	Linear	Logistic	0.467	6.916	1719513	0.199	2.724	0.319	4.814	1088278	0.033	2.695
M3: MLP (5-3-1)	Linear	Tanh	0.440	6.854	1768002	0.478	3.014	0.364	4.649	1064610	0.550	2.845
M4: MLP (5-7-1)	Linear	Exponential	0.443	6.845	1750239	-0.029	2.633	0.380	4.363	1043436	-0.060	3.558
M5: MLP (5-3-1)	Linear	Sine	0.455	6.671	1741698	0.732	2.555	0.373	4.842	1077116	0.755	2.141
M6: MLP (5-5-1)	Logistic	Linear	0.450	6.930	1746433	-0.185	2.862	0.402	4.231	1020800	-0.220	2.370
M7: MLP (5-3-1)	Logistic	Logistic	0.455	6.896	1732843	0.005	2.624	0.320	4.925	1091549	0.306	2.773
M8: MLP (5-4-1)	Logistic	Tanh	0.452	7.137	1774698	-0.243	3.826	0.351	4.634	1068632	-0.402	3.112
M9: MLP (5-10-1)	Logistic	Exponential	0.437	6.916	1771360	0.168	3.134	0.351	4.655	1070552	0.105	4.254
M10: MLP (5-11-1)	Logistic	Sine	0.457	6.734	1750815	0.630	3.136	0.388	4.440	1038758	0.407	2.671
M11: MLP (5-3-1)	Tanh	Linear	0.433	7.037	1768800	0.174	1.920	0.304	5.320	1165426	1.081	2.221
M12: MLP (5-8-1)	Tanh	Logistic	0.464	7.059	1733740	-0.064	3.087	0.263	5.003	1132521	-0.195	3.553
M13: MLP (5-3-1)	Tanh	Tanh	0.522	6.293	1619946	0.219	1.876	0.304	4.822	1124179	0.511	2.175
M14: MLP (5-3-1)	Tanh	Exponential	0.441	7.846	1902359	-2.073	1.352	0.330	4.860	1104011	-0.730	2.686
M15: MLP (5-3-1)	Tanh	Sine	0.513	6.396	1635276	0.280	1.867	0.347	4.570	1086822	0.571	2.205
M16: MLP (5-6-1)	Exponential	Linear	0.444	6.720	1748322	0.186	1.979	0.383	4.374	1046564	0.485	2.904
M17: MLP (5-12-1)	Exponential	Logistic	0.429	7.402	1855749	-0.120	5.806	0.335	5.200	1164488	-0.809	12.309
M18: MLP (5-5-1)	Exponential	Tanh	0.516	6.546	1631085	0.226	1.820	0.276	5.381	1153103	0.699	2.450
M19: MLP (5-4-1)	Exponential	Exponential	0.457	6.847	1725124	-0.022	2.350	0.326	5.001	1095769	0.477	2.464
M20: MLP (5-4-1)	Exponential	Sine	0.394	6.961	1838387	0.446	1.948	0.366	4.527	1095276	0.650	5.819
M21: MLP (5-5-1)	Sine	Linear	0.422	6.995	1789716	0.394	2.992	0.353	4.727	1078469	0.582	3.622
M22: MLP (5-11-1)	Sine	Logistic	0.452	6.831	1761150	0.437	3.365	0.358	4.601	1062896	0.267	3.621
M23: MLP (5-3-1)	Sine	Tanh	0.435	7.087	1766395	0.010	1.873	0.354	4.983	1111610	0.807	1.802
M24: MLP (5-9-1)	Sine	Exponential	0.452	6.849	1740990	-0.078	2.776	0.370	4.473	1053199	-0.128	3.760
M25: MLP (5-4-1)	Sine	Sine	0.452	6.864	1758973	1.187	2.204	0.347	5.331	1170580	1.413	2.110
M26: RBF (5-12-1)	Gaussian	Linear	0.402	7.188	1811467	0.245	2.306	0.197	5.098	1306811	0.395	1.361
M27: MLP-MAF (5-15-1)	Linear, logistic, tanh, exp, sine	Linear	0.443	7.207	1823100	1.753	1.723	0.389	5.403	1261845	2.120	1.911
M28: MLP-MAF (5-12-1)	Logistic, tanh, exp, sine	Linear	0.450	6.747	1736520	0.187	2.171	0.395	4.347	1032719	0.340	2.170
M29: MLP-MAF (5-12-1)	Logistic, tanh, exp	Linear	0.465	6.967	1742633	-0.746	3.119	0.357	4.460	1112156	-0.838	1.785
M30: MLP-MAF (5-12-1)	Linear, sine, cosine	Linear	0.520	6.738	1678592	1.590	1.770	0.481	4.389	961969	-0.310	1.627
			Critical values of *t* and *F*			2.256	1.435	Critical values of *t* and *F*			2.317	2.312

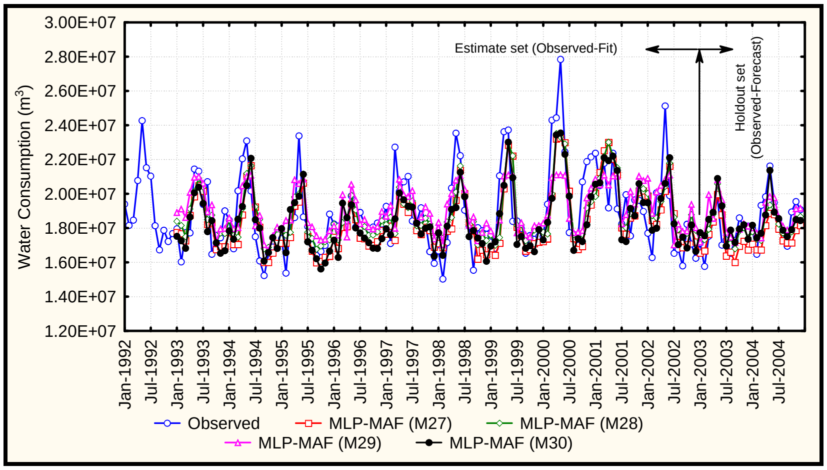

Figure-7. Observed, fit and forecast of MLP-MAF models for monthly water consumption time series of Tampa.

Daily water consumption time series

The second problem that MLP-MAF and traditional MLP and RBF were applied is short term forecasting of daily water consumption time series. The available daily water consumption time series in Tampa is from 1 January 1992 to 31 December 2004 (4745 observations). February 29 in the leap years has been dropped in order to maintain 365 days in each year. The first 4740, 4735 and 4730 observations from 1 January 1992 were used as a training sample for model estimation, and the remaining 5, 10 and 15 observations as holdout sample for forecast evaluation, respectively. Lag correlation coefficients between the present daily demand and past daily values were calculated as shown in Table-2. This information was used to aid in selecting the lagged values as input variables for the developed MLP-MAF and traditional MLP and RBF neural network models. The lagged (past) values used as inputs to the models are the lags 1 through 7 and lag 365, i.e., the past values of water consumption for seven days ago and the consumption of same day for the previous year were used as inputs.

Four MLP-MAF models with different combinations of activation functions in the hidden layer, as described earlier, were developed for daily water consumption time series of Tampa. For comparison purpose, additional forecasts by traditional ANN methods were carried out on the same data. The STATISTICA version 8.0 package software was used to develop twenty five separate traditional MLP models and one RBF model. After applying the models to forecasting daily urban water consumption in the area of Tampa in the USA, the performance of the models were evaluated from the aspect of comparison between the forecasted values and the observed values. The performances of the developed models were evaluated by computing R^2, MAPE and RMSE statistics to estimate set and 5, 10 and 15 days forecasts ahead set. Tables (4) through (6) give the forecasts results for the in-sample (estimate set) and out-sample (holdout set) periods of 5, 10 and 15 days forecasts.

It can be observed that the traditional MLP models performed best than RBF model for the estimate and holdout sets for forecasting 5, 10 and 15 days ahead, with the highest R^2 values of 0.777, 0.769 and 0.685 for models M12, M7 and M21 respectively, and lowest MAPE and MRSE statistics. Comparing traditional MLP models with the developed MLP-MAF models, it is clear that for forecasting 5 days ahead for estimate set there is no significant difference between them. For forecasting 10 and 15 days ahead, the developed MLP-MAF models are better than traditional MLP model with highest R^2 values of 0.775 and 0.758 and lowest MAPE and MRSE values for models M28 and M30 respectively. On the other hand, the statistical analysis on the results of traditional ANN and developed MLP-MAF of 5, 10 and 15 days forecasting ahead models for holdout sets reveals that the MLP-MAF models perform better than traditional MLP and RBF models. Among the four MLP-MAF models developed for each forecasting 5, 10, 15 days ahead, the model M30 which consists of linear, sine and cosine activation functions in the hidden layer and linear activation function in the output layer appears to be the best one with highest R^2 values of 0.814, 0.556, 0.411, lowest MAPE values of 2.224%, 4.301%, 3.698% and lowest RMSE values of 15868, 32479, 31571, respectively. Furthermore, the models M27 and M28 of MLP-MAF appear to have the same forecast performance and be very close to each other for all the daily forecasting (5, 10, 15 days ahead) models. Moreover, it is seen from *t*-test that there is no difference between the means of the observed and forecasted values of estimate and holdout sets.

The accuracy of the models is particularly better for short term forecasting 5 days ahead than 10 days ahead and forecasting 10 days ahead is better than 15 days ahead and this is convenient, because by increasing the forecasting periods ahead the accuracy will decrease. Figures 8 through 10 show the observed and forecasted series of estimate and holdout sets of the four developed

MLP-MAF models for forecasting 5, 10 and 15 days ahead.

As mentioned earlier in monthly water demand forecasting, none of traditional MLP and RBF and MLP-MAF models applied to Tampa daily water consumption time series none of the methods performed very well in forecasting short term urban water demand. This seems to indicate that there were problems with the data used or data pre-processing or the wrong deriving input variables were used. The present models of daily water demand forecasts are nonlinear autoregressive models for time series forecasting by using past water consumptions as inputs to the developed ANN models. The models could potentially be improved if other variables such as socioeconomic and climatic that affect water demand were used. Therefore, a further study recommended to be carried out in the future on the daily water consumption by incorporating both climatic variables and time lagged observations in the input variables of the model.

Table-4. Structure and statistical performance measurements of MLP-MAF, traditional MLP and RBF models of daily water consumption time series for Tampa - forecasting 5 days ahead.

Model	**Hidden layer activation function/s**	**Output layer activation function/s**	**Estimate set** (Observed - Fit)					**Holdout set** (Observed - Forecast)				
			R^2	**MAPE** (%)	**RMSE**	***t*-value**	***F*-value**	**R^2**	**MAPE** (%)	**RMSE**	***t*-value**	***F*-value**
M1: MLP (8-9-1)	Linear	Linear	0.677	4.037	33441	-0.406	1.381	0.614	2.970	19290	0.466	1.238
M2: MLP (8-2-1)	Linear	Logistic	0.741	3.713	29896	-0.173	1.348	0.779	3.485	23576	0.603	2.294
M3: MLP (8-7-1)	Linear	Tanh	0.724	3.789	30986	-0.557	1.447	0.789	3.248	22995	0.679	2.083
M4: MLP (8-10-1)	Linear	Exponential	0.728	3.824	30698	-0.286	1.444	0.810	2.413	18352	0.689	1.407
M5: MLP (8-16-1)	Linear	Sine	0.745	3.681	29742	-0.017	1.250	0.775	3.444	24502	0.751	2.114
M6: MLP (8-11-1)	Logistic	Linear	0.751	3.642	29320	-0.297	1.290	0.643	2.831	22054	0.804	1.214
M7: MLP (8-9-1)	Logistic	Logistic	0.772	3.515	28210	-0.632	1.229	0.653	3.605	23344	0.445	2.081
M8: MLP (8-2-1)	Logistic	Tanh	0.764	3.564	28591	-0.219	1.222	0.578	3.483	24652	0.713	1.596
M9: MLP (8-4-1)	Logistic	Exponential	0.763	3.557	28625	-0.251	1.249	0.758	2.879	20856	0.801	1.387
M10: MLP (8-3-1)	Logistic	Sine	0.752	3.637	29324	-0.282	1.252	0.644	2.848	22727	0.843	1.248
M11: MLP (8-2-1)	Tanh	Linear	0.742	3.698	29863	-0.041	1.282	0.802	3.527	24222	0.595	2.485
M12: MLP (8-7-1)	Tanh	Logistic	0.777	3.478	27847	-0.294	1.195	0.559	4.120	28158	1.013	1.502
M13: MLP (8-2-1)	Tanh	Tanh	0.761	3.581	28880	-0.330	1.179	0.594	3.415	23924	0.636	1.667
M14: MLP (8-2-1)	Tanh	Exponential	0.756	3.647	29129	-0.492	1.263	0.780	2.766	19573	0.642	1.602
M15: MLP (8-13-1)	Tanh	Sine	0.745	3.683	29793	-0.647	1.354	0.780	2.926	19423	0.489	1.865
M16: MLP (8-11-1)	Exponential	Linear	0.749	3.644	29451	0.109	1.381	0.812	2.957	20898	0.661	1.876
M17: MLP (8-5-1)	Exponential	Logistic	0.765	3.573	28594	-0.150	1.178	0.615	3.429	24267	0.866	1.363
M18: MLP (8-13-1)	Exponential	Tanh	0.742	3.693	29924	-0.440	1.419	0.798	3.099	21432	0.610	2.015
M19: MLP (8-6-1)	Exponential	Exponential	0.767	3.519	28403	-0.318	1.257	0.637	3.351	21562	0.768	1.185
M20: MLP (8-17-1)	Exponential	Sine	0.748	3.672	29575	-0.475	1.328	0.785	2.956	19724	0.540	1.843
M21: MLP (8-6-1)	Sine	Linear	0.746	3.684	29699	-0.547	1.347	0.758	2.882	20287	0.649	1.623
M22: MLP (8-9-1)	Sine	Logistic	0.733	3.739	30481	-0.251	1.503	0.753	3.378	24120	0.696	2.104
M23: MLP (8-13-1)	Sine	Tanh	0.725	3.757	30833	0.468	1.395	0.784	3.585	25928	0.734	2.395
M24: MLP (8-2-1)	Sine	Exponential	0.726	3.871	30861	-0.333	1.513	0.812	2.604	18825	0.606	1.671
M25: MLP (8-7-1)	Sine	Sine	0.746	3.675	29683	-0.390	1.389	0.765	2.915	20418	0.640	1.685
M26: RBF (8-17-1)	Gaussian	Linear	0.665	4.221	34092	0.574	1.454	0.772	3.264	22083	0.983	1.200
M27: MLP-MAF (8-20-1)	Linear, logistic, Tanh, Exp, Sine	Linear	0.770	3.519	28243	0.196	1.377	0.625	3.293	23004	0.671	1.565
M28: MLP-MAF (8-16-1)	Logistic, Tanh, Exp, Sine	Linear	0.756	3.614	29085	0.239	1.401	0.729	2.987	21972	0.772	1.541
M29: MLP-MAF (8-12-1)	Logistic, tanh, exp	Linear	0.756	3.612	29071	0.190	1.438	0.803	2.809	19341	0.597	1.738
M30: MLP-MAF (8-12-1)	Linear, Sine, Cosine	Linear	0.758	3.590	29046	-0.634	1.371	0.814	2.224	15868	0.542	1.316
			Critical values of *t* and *F*			2.242	1.061	Critical values of *t* and *F*			2.752	9.605

Table-5. Structure and statistical performance measurements of MLP-MAF, traditional MLP and RBF models of daily water consumption time series for Tampa - forecasting 10 days ahead.

Model	Hidden layer activation function/s	Output layer activation function/s	**Estimate set** (Observed - Fit)					**Holdout set** (Observed - Forecast)				
			R^2	**MAPE** (%)	**RMSE**	***t*-value**	***F*-value**	**R^2**	**MAPE** (%)	**RMSE**	***t*-value**	***F*-value**
M1: MLP (8-4-1)	Linear	Linear	0.753	3.636	29169	0.307	1.367	0.370	4.812	39098	-0.134	1.689
M2: MLP (8-12-1)	Linear	Logistic	0.750	3.647	29376	0.391	1.315	0.377	4.952	39150	-0.128	1.551
M3: MLP (8-4-1)	Linear	Tanh	0.735	3.737	30525	0.397	1.624	0.368	4.948	39438	-0.184	1.606
M4: MLP (8-4-1)	Linear	Exponential	0.733	3.778	30569	0.907	1.306	0.395	4.766	37431	0.019	2.526
M5: MLP (8-4-1)	Linear	Sine	0.751	3.646	29434	0.744	1.408	0.396	4.794	38204	-0.071	1.620
M6: MLP (8-4-1)	Logistic	Linear	0.769	3.539	28205	0.407	1.308	0.393	4.509	38732	-0.153	1.502
M7: MLP (8-16-1)	Logistic	Logistic	0.769	3.555	28228	0.468	1.318	0.369	4.893	39862	-0.173	1.451
M8: MLP (8-6-1)	Logistic	Tanh	0.769	3.574	28247	0.552	1.313	0.343	4.895	40186	-0.082	1.631
M9: MLP (8-16-1)	Logistic	Exponential	0.780	3.470	27509	0.377	1.297	0.379	4.795	39315	-0.224	1.532
M10: MLP (8-8-1)	Logistic	Sine	0.768	3.551	28284	0.363	1.280	0.360	4.769	39889	-0.142	1.534
M11: MLP (8-9-1)	Tanh	Linear	0.757	3.613	28987	0.508	1.360	0.383	4.621	38785	-0.133	1.611
M12: MLP (8-12-1)	Tanh	Logistic	0.766	3.593	28450	0.550	1.310	0.361	4.821	39614	-0.130	1.610
M13: MLP (8-8-1)	Tanh	Tanh	0.771	3.542	28084	0.380	1.339	0.341	4.967	40301	-0.133	1.637
M14: MLP (8-4-1)	Tanh	Exponential	0.777	3.491	27711	0.351	1.262	0.371	4.934	39522	-0.137	1.507
M15: MLP (8-9-1)	Tanh	Sine	0.769	3.545	28175	0.261	1.284	0.360	4.809	39901	-0.193	1.562
M16: MLP (8-7-1)	Exponential	Linear	0.765	3.563	28498	0.500	1.299	0.384	4.593	38754	-0.076	1.579
M17: MLP (8-10-1)	Exponential	Logistic	0.759	3.615	28861	0.443	1.329	0.368	4.802	39216	-0.169	1.684
M18: MLP (8-11-1)	Exponential	Tanh	0.767	3.561	28318	0.336	1.303	0.372	4.529	39317	-0.135	1.570
M19: MLP (8-12-1)	Exponential	Exponential	0.768	3.551	28312	0.423	1.291	0.371	4.825	39462	-0.111	1.526
M20: MLP (8-10-1)	Exponential	Sine	0.768	3.538	28261	0.315	1.283	0.381	4.663	38917	-0.151	1.589
M21: MLP (8-6-1)	Sine	Linear	0.750	3.655	29368	0.482	1.369	0.383	4.645	38686	-0.068	1.613
M22: MLP (8-4-1)	Sine	Logistic	0.747	3.637	29508	0.267	1.275	0.386	5.027	38912	-0.159	1.536
M23: MLP (8-9-1)	Sine	Tanh	0.733	3.697	30517	0.620	1.514	0.407	4.597	38441	-0.178	1.437
M24: MLP (8-12-1)	Sine	Exponential	0.722	3.831	31111	0.527	1.256	0.415	4.697	37019	-0.108	1.966
M25: MLP (8-13-1)	Sine	Sine	0.751	3.640	29458	0.671	1.432	0.398	4.664	37811	-0.093	1.798
M26: RBF (8-3-1)	Gaussian	Linear	0.632	4.425	35628	0.550	1.560	0.125	6.047	45107	0.010	5.994
M27: MLP-MAF (8-20-1)	Linear, Logistic, Tanh, Exp, Sine	Linear	0.772	3.519	28084	0.462	1.312	0.384	4.629	38864	-0.126	1.558
M28: MLP-MAF (8-16-1)	Logistic, Tanh, Exp, Sine	Linear	0.775	3.492	27893	0.490	1.280	0.383	4.608	38825	-0.079	1.565
M29: MLP-MAF (8-12-1)	Logistic, Tanh, Exp	Linear	0.753	3.636	29169	0.307	1.367	0.370	4.812	39098	-0.134	1.689
M30: MLP-MAF (8-12-1)	Linear, Sine, Cosine	Linear	0.744	3.795	29699	-0.042	1.252	0.556	4.301	32479	-0.253	1.786
			Critical values of *t* and *F*			2.242	1.061	Critical values of *t* and *F*			2.445	4.026

Table-6. Structure and statistical performance measurements of MLP-MAF, traditional MLP and RBF models of daily water consumption time series for Tampa - forecasting 15 days ahead.

Model	Hidden layer activation function/s	Output layer activation function/s	**Estimate set** (Observed - Fit)					**Holdout set** (Observed - Forecast)				
			R^2	MAPE (%)	RMSE	*t*-value	*F*-value	R^2	MAPE (%)	RMSE	*t*-value	*F*-value
M1: MLP (8-5-1)	Linear	Linear	0.673	4.135	33886	-0.464	1.375	0.305	4.110	34427	-0.511	3.278
M2: MLP (8-4-1)	Linear	Logistic	0.671	4.152	33920	0.331	1.398	0.297	4.120	34084	-0.069	4.008
M3: MLP (8-4-1)	Linear	Tanh	0.671	4.164	34047	-0.693	1.355	0.312	4.120	34747	-0.684	2.900
M4: MLP (8-6-1)	Linear	Exponential	0.658	4.456	35786	0.988	2.275	0.293	4.133	34966	0.101	7.964
M5: MLP (8-4-1)	Linear	Sine	0.672	4.140	33898	-0.464	1.397	0.305	4.102	34438	-0.518	3.347
M6: MLP (8-12-1)	Logistic	Linear	0.680	4.048	33428	-0.303	1.414	0.327	4.076	33708	-0.373	2.509
M7: MLP (8-6-1)	Logistic	Logistic	0.670	4.184	34163	-0.799	1.332	0.301	4.224	35220	-0.689	2.509
M8: MLP (8-7-1)	Logistic	Tanh	0.674	4.148	33992	-0.667	1.301	0.307	4.160	34801	-0.628	2.650
M9: MLP (8-9-1)	Logistic	Exponential	0.680	4.068	33427	-0.180	1.372	0.318	4.098	33992	-0.435	2.818
M10: MLP (8-11-1)	Logistic	Sine	0.669	4.159	34221	-0.669	1.325	0.315	4.112	34478	-0.583	2.615
M11: MLP (8-8-1)	Tanh	Linear	0.672	4.127	34000	-0.598	1.334	0.314	4.112	34377	-0.554	2.790
M12: MLP (8-7-1)	Tanh	Logistic	0.677	4.114	33792	-0.604	1.305	0.303	4.175	34806	-0.598	2.675
M13: MLP (8-5-1)	Tanh	Tanh	0.674	4.135	33840	-0.647	1.419	0.320	4.037	34259	-0.581	2.933
M14: MLP (8-9-1)	Tanh	Exponential	0.673	4.145	34096	-0.785	1.301	0.310	4.162	34978	-0.698	2.571
M15: MLP (8-7-1)	Tanh	Sine	0.676	4.114	33818	-0.681	1.332	0.310	4.142	34654	-0.615	2.734
M16: MLP (8-9-1)	Exponential	Linear	0.679	4.063	33533	-0.446	1.401	0.314	4.090	34029	-0.415	2.858
M17: MLP (8-13-1)	Exponential	Logistic	0.673	4.126	34018	-0.593	1.300	0.315	4.133	34331	-0.523	2.582
M18: MLP (8-10-1)	Exponential	Tanh	0.675	4.086	33768	-0.242	1.326	0.309	4.135	34068	-0.356	2.812
M19: MLP (8-7-1)	Exponential	Exponential	0.678	4.072	33696	-0.512	1.318	0.319	4.120	34084	-0.457	2.536
M20: MLP (8-6-1)	Exponential	Sine	0.678	4.068	33584	-0.546	1.426	0.327	4.056	33855	-0.448	2.558
M21: MLP (8-6-1)	Sine	Linear	0.685	4.038	33160	-0.169	1.356	0.317	4.107	33885	-0.389	2.991
M22: MLP (8-9-1)	Sine	Logistic	0.682	4.160	33662	0.718	1.742	0.312	4.057	33874	-0.007	4.658
M23: MLP (8-5-1)	Sine	Tanh	0.671	4.191	34193	-1.100	1.406	0.308	4.104	35233	-0.801	2.999
M24: MLP (8-11-1)	Sine	Exponential	0.664	4.413	35460	0.750	2.259	0.295	4.056	34863	0.040	7.696
M25: MLP (8-7-1)	Sine	Sine	0.680	4.085	33498	-0.480	1.390	0.312	4.095	34248	-0.508	3.179
M26: RBF (8-13-1)	Gaussian	Linear	0.677	4.181	34358	-0.855	1.153	0.267	4.336	35454	-0.462	2.672
M27: MLP-MAF (8-20-1)	Linear, Logistic, Tanh, Exp, Sine	Linear	0.674	4.099	33830	-0.395	1.339	0.314	4.114	34138	-0.457	2.772
M28: MLP-MAF (8-16-1)	Logistic, Tanh, Exp, Sine	Linear	0.675	4.096	33811	-0.430	1.337	0.315	4.113	34139	-0.468	2.738
M29: MLP-MAF (8-12-1)	Logistic, Tanh, Exp	Linear	0.674	4.166	34358	-0.901	1.194	0.307	4.224	35172	-0.705	2.410
M30: MLP-MAF (8-12-1)	Linear, Sine, Cosine	Linear	0.758	3.612	29029	-0.034	1.294	0.411	3.698	31571	-0.247	1.842
			Critical values of *t* and *F*			2.242	1.061	Critical values of *t* and *F*			2.368	2.979

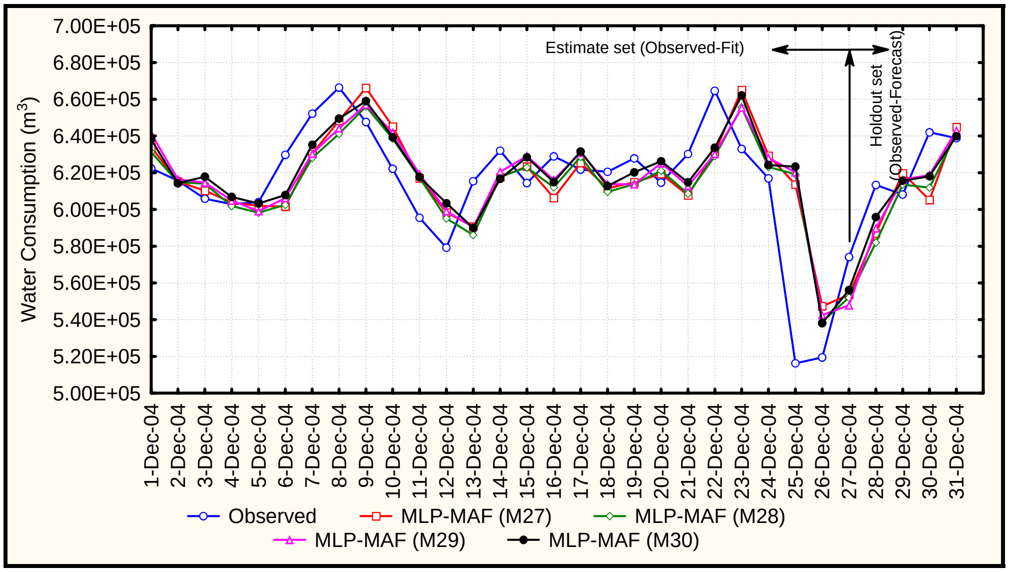

Figure-8. Observed, fit and forecast (5 days ahead) of MLP-MAF models for daily water consumption time series of Tampa.

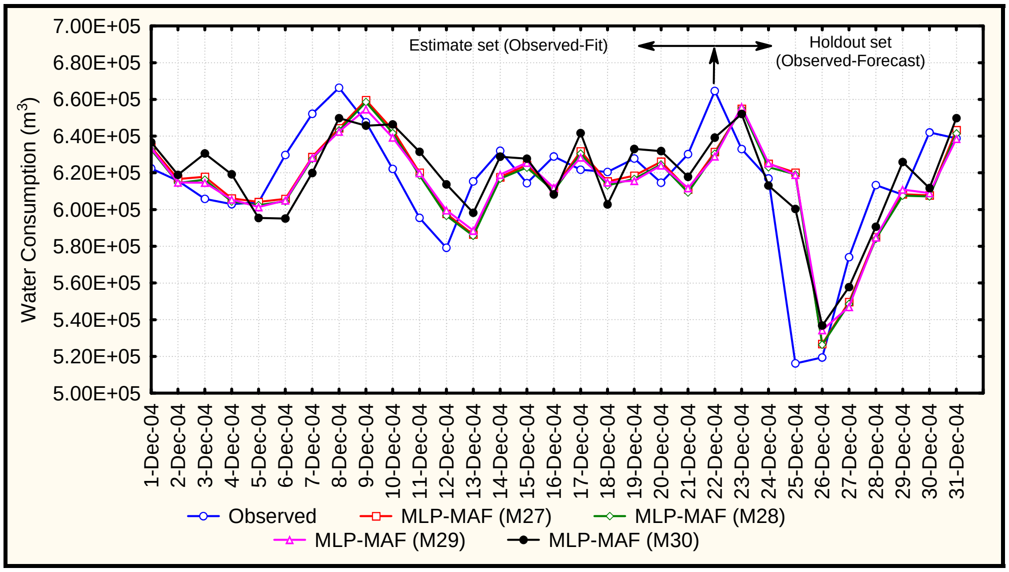

Figure-9. Observed, fit and forecast (10 days ahead) of MLP-MAF models for daily water consumption time series of Tampa.

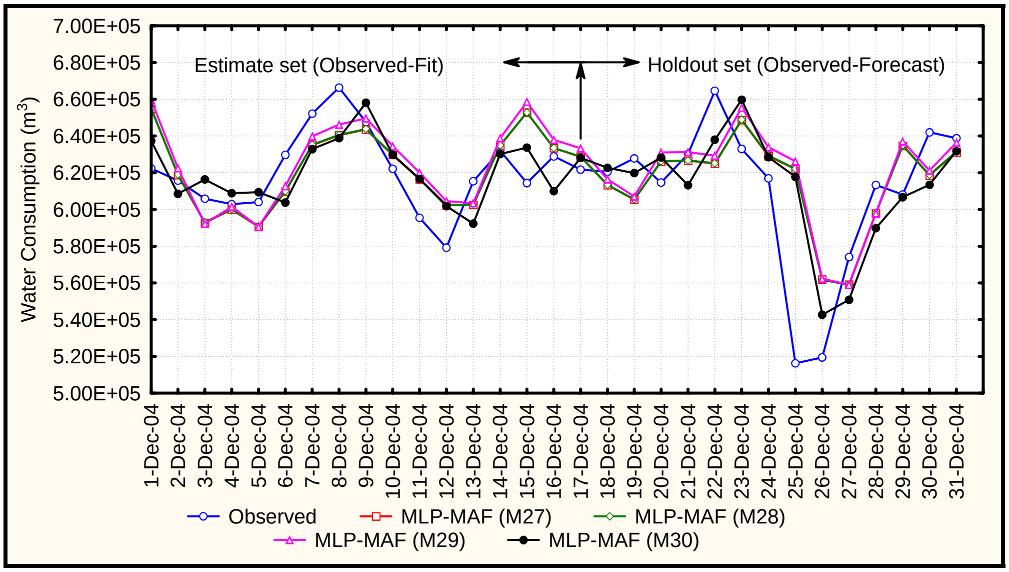

Figure-10. Observed, fit and forecast (15 days ahead) of MLP-MAF models for daily water consumption time series of Tampa.

CONCLUSIONS

The model developed in this research for daily and monthly municipal water demand forecasting is a Multilayer Perceptron - Multiactivation Function (MLP-MAF) neural network that contains different activation functions in the nodes of the hidden layer. Even though the developed MLP-MAF time series models for water consumption time series of Tampa had not performed well, the models are better than traditional MLP and RBF neural network models in terms of statistical measurements, R^2, MAPE and RMSE. Among the different activation functions in the hidden layer of MLP-MAF, the combination of linear, sine and cosine functions is performing better than other combinations for daily and monthly water consumption time series of Tampa. Comparing the results of the traditional MLP and RBF neural network models for both daily and monthly water demand forecasting indicates that the forecasting accuracy of traditional MLP model is more accurate than the traditional RBF model.

Appendix-I: Notations

Symbol	Description
b	The bias to the neuron in a layer of neural network
F	The classical F statistic
$f_h(.)$	The activation function of hidden layer in neural network
f_k	The activation function of neuron in a layer of neural network
$f_o(.)$	The activation function of output layer
f_p	The activation function of group p in the hidden layer of MLP-MAF
g	The number of neuron groups in the hidden layer of MLP-MAF
HIN	The inputs to the hidden layer of MLP-MAF
I	The inputs to the input layer of MLP-MAF
N	The number of data points
n	The number of neurons in a neural network
O	The outputs of the output layer of MLP-MAF
S_d	The standard deviation
$S_y^2, S_{\hat{y}}^2$	The variance of observed and forecasted data, respectively
t	The time (day, month)
t_c	The critical t-statistic at a significance level
$t - statistic$	The classical t-statistic
w	The weight between the layers of MLP network
WI	The input-hidden layers weights of MLP-MAF
WO	The hidden-output layers weights of MLP-MAF
x	The independent variable
$\bar{X}$	The mean of time series X_t
X_i	The linear regression value for annual means
x_{max}	The maximum value of series x
x_{mean}	The mean value of series x
x_{min}	The minimum value of series x

x_{ni}	The scaled value of variable x_i
x_{sd}	The standard deviation of series x
y	The dependent variable
y_k	The output of neuron k in a layer of neural network
$\hat{y}_k$	Output of a neural network model
y_t	The observation at time t
$\hat{y}, \tilde{y}$	The forecasted data with its mean for observed data y_i
$y_i, \bar{y}$	The observed data with its mean, respectively
ε_t	The random error
μ	The parameters in the Gaussian function
η_a	The learning rate for input-hidden weights of MLP-MAF
η_b	The learning rate for hidden-output weights of MLP-MAF
σ	The standard deviations in the Gaussian function

Appendix-II: Abbreviations

Abbreviation	Description
ANN	Artificial neural network
CE	Coefficient of efficiency
DAN2	Dynamic artificial neural network model
FFBP	Feed forward back propagation
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MLP	Multilayer perceptron
MSE	Mean squared error
R^2	Coefficient of determination
RBF	Radial basis function
RMSE	Root mean squared error
SSE	Sum of square error
SVM	Support vector machines
USA	The United State of America
MLP-MAF	Multilayer perceptron multiactivation function neural network

REFERENCES

Abraham A. 2005. Artificial Neural Networks. In: Handbook of Measuring System Design. s.l.: John Wiley and Sons, Ltd. pp. 901-908.

Adamowski J. F. 2008. Peak Daily Water Demand Forecast Modelling Using Artificial Neural Networks. Journal of Water Resources Planning and Management. 134(2): 119-128.

Alaa H. A. and Nisai W. 2004. Short Term Forecasting for Urban Water Consumption. Journal of Water Resources Planning and Management, 1 September. 130(5): 405-410.

Asefa T. and Adams A. 2007. Short Term Urban Water Demand Forecast Models in Action: Challenges from Model Development to Implementation to Real Time Operations. Tampa, Florida, ASCE.

Bougadis J., Adamowski K. and Diduch R. 2005. Short Term Municipal Water Demand Forecasting. Journal of Hydrological Processes, January. 19(1): 137-148.

Chang N. and Makkeasorn A. 2006. Water Demand Analysis in Urban Region by Neural Network Models. Cincinnati, Ohio, USA, ASCE.

Dreyfus G. 2005. Neural Networks: Methodology and Applications. 2nd Ed. s.l.:Springer-Verlag Berlin Heidelberg.

Engelbrecht A. P. 2007. Computational Intelligence: An Introduction. 2nd Ed. The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ: John Wiley and Sons Ltd.

Firat M. and Gungor M. 2007. River Flow Estimation Using Feed Forward and Radial Basis Neural Networks Approaches. Antalya, Turkey, INPIM. pp. 599-611.

Firat M., Yurdusev M. A. and Turan M. E. 2009. Evaluation of Artificial Neural Network Techniques for Municipal Water Consumption Modelling. Journal of Water Resources Management. 23(4): 617-632.

Ghiassi M., Zimbra D. K. and Saidane H. 2008. Urban Water Demand Forecasting with a Dynamic Artificial Neural Network Model. Journal of Water Resources Planning and Management. 134(2): 138-146.

Hill T. and Lewicki P. 2007. Statistics Methods and Applications. Tulsa, OK: StatSoft, Inc.

Jain D. A., Varshney A. K. and Joshi U. C. 2001. Short Term Water Demand Forecasting at IIT Kanpur Using Artificial Neural Networks. Journal of Water Resources Management. 15(5): 299-321.

Kim J. H., Hwang S. H. and Shin H. S. 2001. A Neuro-Genetic Approach for Daily Water Demand Forecasting. KSCE. Journal of Civil Engineering, September. 5(3): 281-288.

Kingston G. B. 2006. Ph.D Thesis. Adelaide: Computer and Mathematical Sciences, Faculty of Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Australia.

Leon F. and Zaharia M. h. 2010. Stacked Heterogeneous Neural Network for Time Series Forecasting. Mathematical Problems in Engineering. pp. 1-21.

Makridakis S., Wheelwright S. C. and Hyndman R. J. 1998. Forecasting, Methods and Applications. 3rd Ed. New York: John Wiley and Sons, Inc.

Msiza I. S., Nelwamondo F. V. and Marwala T. 2007. Artificial Neural Networks and Support Vector Machines for Water Demand Time Series Forecasting. Montreal, Quebec, Canada, IEEE. pp. 638-643.

Msiza I. S., Nelwamondo F. V. and Marwala T. 2007. Water Demand Forecasting Using Multi-Layer Perceptron and Radial Basis Functions. Orlando, Florida, USA, IEEE. pp. 13-18.

Palit A. K. and Popovic D. 2005. Computational Intelligence in Time Series Forecasting. Theory and Engineering Applications. London: Springer-Verlag London Limited.

Rumantir G. W. 1995. M.Sc Thesis. Melbourne: Department of Computer Science. Royal Melbourne Institute of Technology.

Samarasinghe S. 2006. Nueral Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition. New York: Taylor and Francis Group, LLC.

Taylor J. G. 1996. Neural Networks and Their Applications. Baffins Lane, Chichester, West Sussex PO19 1UD: John Wiley and Sons, Ltd.

Zhang G., Patuwo B. E. and Hu M. Y. 1998. Forecasting With Aartificial Neural Networks: The State of the Art. International Journal of Forecasting. (14): 35-62.

Zhang G. P., Patuwo B. E. and Hu M. Y. 2001. A Simulation Study of Artificial Neural Networks for Nonlinear Time Series Forecasting. Journal of Computers and Operations Research, April. 28(4): 381-396.

Zhang J., Song R., Bhaskar N. R. and French M. N. 2006. Short Term Water Demand Forecasting: A Case Study. Cincinnati, Ohio, USA, ASCE.

Zhang J., Song R., Bhaskar N. R. and French M. N. 2007. Ensemble Forecasting of Daily Water Demand. Tampa, Florida, ASCE.