Forecasting Nile River Flood Using a Fuzzy Neural Network Model

Azza El-Naggar[1] Eman Sanhouri[2]

Abstract

River flood forecasting has a significant social and economic impact as it can help in protection from water shortages and possible flood damages. River flood forecasting is a very difficult process since it means handling large amounts of dynamic nonlinear systems with a great amount of uncertainty and noisy data. In addition, the data about many variables that affect flood are not available and the underlying physical relationships are not fully understood. A model that combines both neural networks and fuzzy systems can be effective in handling this problem. This system will have the ability to learn from data with good generalization capability using neural networks and to deal effectively with uncertainty using fuzzy systems. ANFIS (Adaptive Network based Fuzzy Inference System) is a model that combines both neural networks and fuzzy systems. In this paper we use ANFIS for forecasting river Nile flood. In addition to ANFIS, we use regression and neural networks for river Nile flood forecasting and then we make a comparison between the performances of the three techniques in forecasting river Nile flood.

Keywords: Fuzzy systems, neural networks, ANFIS, neural fuzzy systems, Nile River, flood forecasting, subtractive clustering, fuzzy clustering, preprocessing data techniques.

1. Introduction

The River Nile is the main source of life to many African countries in general and to Egypt in particular. Egypt is the most downstream country and basically depends on the river Nile for getting water. The climate is arid and annual rainfall is too little to depend on it as a source of water. Agriculture in Egypt is possible mainly with irrigation. Forecasting the river flow has a significant economic impact, as it helps in management of agricultural water, prediction of water storage, protection from potential flood damage, and estimation of bridges load. Forecasting River Nile flow can also help in determining the optimum amount of water to be released by high dam, and thus can help to more efficient management of the water [2], [17].

The river flood forecasting problem has been traditionally tackled using techniques such as regression, neural networks and fuzzy systems. Previous studies show that flood has a great amount of uncertainty; many variables that affect flood are not taken in consideration because the data about them are not available and even if they are available their exact relationship with flood is not clear. River flood forecasting needs new techniques that can deal effectively with uncertainty, have the ability to generalize to unseen patterns and in the same time do not need a mathematical representation of the flood process.

---

[1]Azza El-Naggar: Assistant Professor – Faculty of Economics and Political Science – Cairo University
[2]Eman Sanhouri: Instructor – Faculty of Commerce and Business Administration – Helwan University

Neural networks have the ability to learn from training data with good generalization capability. Fuzzy set techniques are known as effective tools in dealing with systems that have great amount of uncertainty and strongly nonlinear behavior of time varying characteristics. Both techniques are nonlinear models that do not need a mathematical formulation of the underlying system. It seams that combining both techniques can provide better performance than that obtained from the previous mentioned techniques.

One of the most powerful models that combine both neural networks and fuzzy set techniques is ANFIS (Adaptive Network based Fuzzy Inference System). ANFIS is known for having a great capability in approximating complex nonlinear models with high degree of accuracy. In this paper, we use ANFIS for forecasting river Nile flood and then we make a comparison with regression and neural network techniques in solving this problem.

2. Review of Some Models Used In River Flow Forecasting

There are different techniques used in forecasting river flow. Examples of these techniques are regression analysis, neural network and fuzzy systems. We review in this section four previous models that were used for forecasting river flow: two for the river Nile and two for other rivers.

The first model for the river Nile is a multivariate step-wise linear regression model, used for real-time forecasting and for generating multi-lead forecasts for monthly discharges at different stations on the Nile Basin [8], [19]. The model uses past and present discharges of the station at which the forecasting is done and also at the upstream stations. This model has the advantages of a small sum of square errors and of being a multivariate model, yet it has the disadvantages of tending to either overestimate low floods or underestimate high floods. In other words, the performance of the model is excellent for low or medium flows. Since the main advantage of forecasting a river flow is forecasting the high flows (flood) to prevent damage resulting from them, then there is a need for another method to overcome these disadvantages [8], [19].

The second model for the river Nile developed by Atiya et al. [2], used the neural network for the river Nile flow forecasting. They used readings of the flow volume for each ten-days and month period at Dongola station located in northern Sudan (south of high dam). The data spanned the period from 1974-1992. They used a network consisting of one hidden layer with three hidden nodes and trained it using the standard back-propagation method. The Normalized Root Mean Square Error (NRMSE) is used as a measurement of error. Different combinations of inputs were used and they reported that combinations possessing the flow value at the same period to be forecasted but one year ago resulted in lower error than combinations possessing purely previous flow value. The network has reasonable generalization ability, and quite accurate forecasts for all periods of the year were obtained except for the peak flow periods (flood periods) where there were some small errors. So they suggested training a separate network for the high flow periods with some kind of expert type network.

See et al. [18], used a neural network for real-time flood forecasting for the Ouse River catchments in the United Kingdom. All data were originally recorded at 15 minute intervals. However to reduce the amount of data, hourly averages were used. Five years of hourly water level data (1982- 1986) with the previous five hourly observations as inputs were used to fit both ARMA (Auto-Regressive Moving Average) models and Neural Networks. Three years of testing data (1987 - 1989) were then used to validate the model performance. The study aimed to make horizon forecast of six hours and used Root Mean Square Error (RMSE) as a measurement of errors. They found that as a global model, neural network performs worse than statistical models on the important flood events but by simply disaggregating the data into low and high level events, the neural networks can concentrate their efforts on learning a smaller number of similar patterns and thus significantly improve their forecasting accuracy. They recommended experiments with more intelligent disaggregating techniques such as fuzzy logic, where rainfall data might have been clustered by previous level measurements with a self organizing neural network.

Stuber et al. [19], used fuzzy model for forecasting a six hour flow level at river Trier/Mosel in Germany. They used only discharges as input variables, no rainfall data or other variables were considered. Measured data were used from nine previous flood events for the generation of the model, and then the model was tested with four other events. Data were measured and provided regularly in one-hour periods. They used Takagi-Sugeno model and assigned algebraic product for AND operator. Simulation results show that at least the same forecast quality (compared with other techniques) was achieved even without the consideration of measured rainfall values and rainfall forecasts.

3. Adaptive Network based Fuzzy Inference System (ANFIS)

Although fuzzy systems and Neural Networks have many advantages, they have drawbacks that may limit their usage as effective nonlinear and time variant solving techniques, [13], [15]. Combining Neural Network and fuzzy logic system can avoid many disadvantages and gain many advantages to both techniques. From the point view of fuzzy systems, the most important reason for combining fuzzy systems with neural networks is their learning capability. Combining neural networks with fuzzy systems enhances the generalization capabilities of fuzzy systems and makes it easier to generate fuzzy systems that achieve a pre-specified accuracy output. From the point view of neural networks, such combinations enables neural network to incorporate prior knowledge into the system, which shortens the learning process. In addition, the structure of neural networks will be determined by the rules and fuzzy sets used for the underlying problem such that there is no need to specify any network parameters (e.g. number of hidden units). Using fuzzy logic System with neural network can speed up convergence of neural networks by using variable learning rates.

ANFIS is one of the first models that combine fuzzy systems and neural networks. It was proposed by Jang in 1992 as stated by Rajasekaran [16]. ANFIS gained a great fame as an effective technique in dealing with complex nonlinear systems since it approximates these systems with a high degree of accuracy. It can generate fuzzy systems from input-output data automatically using feed-forward neural networks. ANFIS is a neural fuzzy system with Takagi-Sugeno fuzzy rule whose consequents are linear combinations of their preconditions. The Takagi-Sugeno fuzzy rules have the following form [13]:

R^j: IF x_1 is A_1^j AND x_2 is A_2^j .. AND x_n is A_n^j THEN $y = f_j = a_o^j + a_1^j x_1^j .. + a_n^j x_n^j$ (1)

where x_i is an input variable, y is the output variable, A_i^j 's are linguistic terms of the precondition part, with membership function $\mu_{A_i^j}(x_i)$, $a_i^j \in \Re$ are the coefficients of the linear equations $f_j(x_1, .., x_n)$ and $j=1, ..., m$, $i=1, ..., n$. For simplicity, assume that the fuzzy system under consideration has two inputs (x_1 and x_2) and one output y and that the rule base contains two Takagi-Sugeno fuzzy rules as follows:

R^1: IF x_1 is A_1^1 AND x_2 is A_2^1 THEN $y = f_1 = a_o^1 + a_1^1 x_1^1 + a_2^1 x_2^1$

R^2: IF x_1 is A_1^2 AND x_2 is A_2^2 THEN $y = f_2 = a_o^2 + a_1^2 x_1^2 + a_2^2 x_2^2$ (2)

For the given input values x_1 and x_2 the inferred output Y^* is calculated by

$$Y^* = \overline{w}_1 f_1 + \overline{w}_2 f_2 \quad (3)$$

where $\overline{w}_j$ is the firing strength of R^j, j =1, 2 relative to the sum of all rules firing strength. The corresponding ANFIS structure is shown in Figure 1 [13].

ANFIS consists of 6 layers, each node in these layers functions as follows:

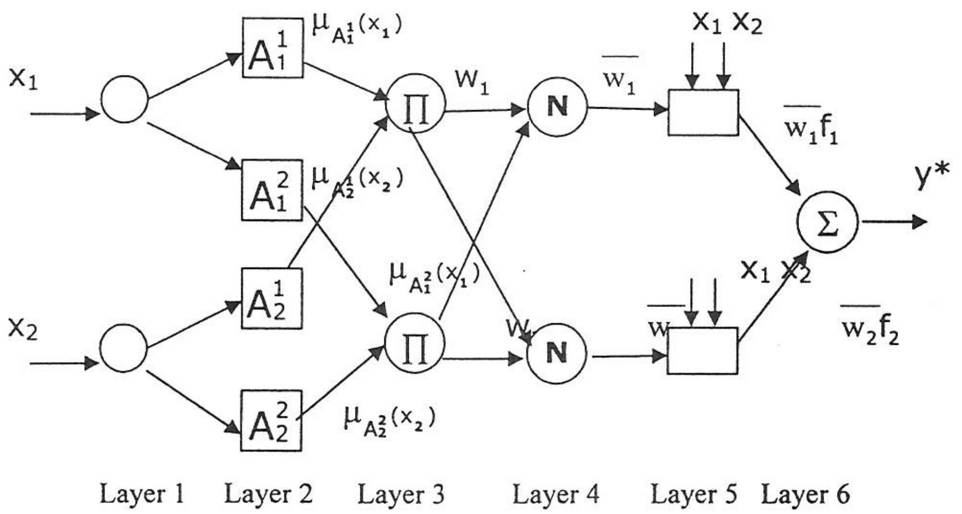

Figure 1: Structure of ANFIS.

Layer 1: Every node in this layer is an input node that just passes external signals to the next layer.

Layer 2: Every node in this layer acts as membership function $\mu_{A_i^j}(x_i)$, and its output specifies the degree to which the given x_i satisfies the quantifier A_i^j. Usually, we choose $\mu_{A_i^j}(x_i)$ to be bell-shaped with 1 as a maximum and 0 as a minimum, such as

$$\mu_{A_i^j}(x_i) = \frac{1}{1 + \{[(x_i - m_i^j) / \sigma_i^j]^2\}^{b_i^j}} \quad (4)$$

or

$$\mu_{A_i^j}(x_i) = \exp\left[-\left(\frac{x_i - m_i^j}{\sigma_i^j}\right)^2\right]^{b_i^j} \tag{5}$$

where $\{m_i, \sigma_i, b_i^j\}$ is the parameter set to be tuned. In fact, any continuous and piece-wise-differentiable functions, such as the commonly used Gaussian or the triangular membership functions, are also qualified candidates for node functions in this layer. Parameters in this layer are referred to as *precondition or antecedent parameters.*

Layer 3: Every node in this layer is a fixed node labeled $\prod$. It multiplies incoming signals to obtain w_j, i.e. $w_j = \prod_i \mu_{A_i^j}(x_i)$, and sends the product out. Each node output represents the firing strength of a rule. In fact, other t-norm operators, other than the product, can also be used as the node function for the generalized AND function [10].

Layer 4: Every node in this layer is a fixed node labeled N. The jth node calculates the ratio of the jth rule's firing strength to the sum of all the rules firing strengths

$$\overline{W_j} = \frac{W_j}{\sum_{j=1}^{m} W_j} \quad , j=1,2,\ldots,m. \tag{6}$$

Layer 5: Every node j in this layer calculates the weighted consequent value $\overline{W_j} f_j$, where $\overline{W_j}$ is the output of layer 4 and (a_0^j, a_1^j, a_2^j) is the parameter set to be tuned. Parameters in this layer are referred to as *consequent parameters.*

Layer 6: The only node in this layer labeled Σ, and it sums all incoming signals to obtain the final inferred result Y^* for the whole system.

ANFIS is functionally equivalent to fuzzy systems with Takagi-Sugeno rules. It updates its parameters according to back-propagation algorithm. Also, ANFIS can make use of a mixture of back-propagation to adapt parameters of the membership functions and uses the least mean squares estimation procedure to determine the coefficients of the linear combination in the rule's consequents. This procedure is called *hybrid learning*. A step in this learning procedure has two parts. In the first part, (called *forward pass*) the input patterns are propagated, and the optimal consequent parameters are estimated by iterative least squares procedure, while the antecedent parameters (membership functions) are assumed to be fixed for the current cycle through the training set. In the second part (called *backward pass*), the patterns are propagated again, and in this epoch, back-propagation is used to modify the antecedent parameters, while the conclusion parameters remain fixed [1]. Hybrid learning is known to yield better results than those obtained by using back-propagation only.

ANFIS needs an initial fuzzy system. This initial fuzzy system can be created using input space partitioning techniques. There are many techniques that can be used in input space partitioning. The most famous one is *grid partitioning method* [4] in which the domains of antecedent variables are a priori partitioned by a number of membership functions. They are usually evenly spaced and shaped. The rule base is then established to cover all the combinations of the antecedent terms. A severe drawback of this approach is that the number of rules grows exponentially since it equals the product of the number of fuzzy partitions per each input variable. This problem is known as the *curse of dimensionality problem* which means that, the number of parameters that need to be optimized is large relevant to the available data set.

To overcome this problem either crisp or fuzzy clustering methods can be used. Using cluster methods minimizes the rule base size and so help in solving the curse of dimensionality problem. In crisp clustering techniques we can find objects that can not be assigned strictly to one cluster because they are located between clusters. In addition, crisp clustering techniques can not handle uncertain and noisy data. In these cases it is preferable to use fuzzy clustering techniques since they can effectively handle noisy and uncertain data [6]. Fuzzy clustering techniques use the concept of fuzzy membership to some prototypical object. The degree of similarity can be calculated using a suitable distance measure in such a way that the objects within each cluster are as similar as possible, and the objects in different clusters are dissimilar as much as possible [5]. The antecedent membership functions are then extracted by projecting the clusters onto the individual variables. The number of clusters in the data can either be determined a priori such as fuzzy c-means clustering method (FCM) or sought automatically such as subtractive clustering method.

ANFIS is a local model approach as it uses Takagi-Sugeno fuzzy rule. The Takagi-Sugeno fuzzy rule functions as a piece-wise linear approximation of a nonlinear function (see Figure 2). In local models, the structure depends on decomposition of the considered input space into different operating regimes. Within each operating regime, a simple local sub-model is valid (each sub-model is valid in a specific region). The overall model output is given by the combination of all locally active sub-models [3]. Local models can deal with systems that have a high degree of complexity and nonlinearity where a global model may be not convenient.

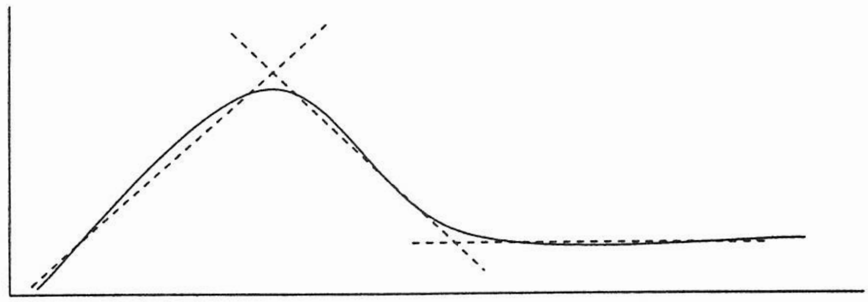

Figure 2: Takagi-Sugeno model as smoothed piece-wise linear approximation of a nonlinear function.

4. The Application

4.1 Data Collecting and Handling

Nile river flow depends on a large number of variables. These variables include meteorological, physical, and time variables. However, since data about most of these variables are not available or available for short or intermitted periods we will use only discharges of Aswan station and upstream stations to predict Nile flood flow at Aswan station. We choose stations that lie on the main rivers that provide Egypt's Nile river with its water to limit the number of stations (variables) taken in consideration. For each main river we choose only one station to avoid the multi-collinearity problem from which the multivariate step-wise regression model suffers [19]. A station is chosen such that it lies at the end of the river, so that all tributaries of this river are taken in consideration. Accordingly, we collected data from four stations in the period (1912-2000). So we have 89 pairs of input-output data. These stations are Malakal, Khartoum, Atbara and Aswan. Since we are interested in forecasting monthly flood flows, the obtained stations' discharges are the used monthly discharges. The used model (ANFIS) is known to require enough data set size for the training data set to give good results. We choose to train the model with 80 pairs of input-output data set and test the model with the remaining data set (9 pairs of input-output data).

The collected data are not of the same nature; that is some of them are natural discharges and some are not. By natural discharge, we mean the discharge as it is given by nature [19] without any losses in the river discharge that can happen by nature such as evaporation or by man such as reserve a quantity of water using dams (for example, High dam). So data are handled such that they are all natural discharges. This process is called naturalization process. Equations that convert discharge to natural discharge are taken from specialists on river Nile (Engineer Ahmed Fahmy at the Ministry of Water Resources and Irrigation). The data of Aswan and Malakal are given after naturalization, so we naturalize the discharges of Atbara and Khartoum as follows:

Natural discharge at Atbara = discharge at Atbara + Release by Khashm el-Kerba canal + Evaporation at Khashm el-Kerba reservoir + change in contents of Khashm el-Kerba reservoir

Natural discharge at Khartoum = discharge at Khartoum + discharge at Gezira and el-Manakel + evaporation at Sennar station+ change in contents of Sennar station + evaporation at Roseirs station + change in contents of Roseirs station

After naturalization, we perform some data preprocessing techniques. Data preprocessing is to remove irrelevant information and extract key features of the data to simplify solving the problem without throwing any important information [12]. The objectives of data preprocessing is size reduction of the input space, smoother relationship, noise reduction, and feature extraction. One method of feature extraction is data normalization. Data normalization (scaling) can provide a better model and avoid numerical problems. Normalization is a linear scaling algorithm. It transforms the original inputs range into a new data range (in most cases interval [0, 1] is chosen). One of the most important and used techniques of data normalization is *Min-Max method* is given as

$$Y_{new} = \left(\frac{Y_{old} - \min_1}{\max_1 - \min_1} \right) \left(\max_2 - \min_2 \right) + \min_2 \tag{7}$$

Table 1: ANFIS related information.

	Variables Used in Prediction	Number of rules	Number of parameters	Cluster Radius	Learning rates
July	June-Atbara June-Malakal June-Aswan	7	70	.30	.10
August	July-Atbara July-Khartoum July-Malakal	4	40	.45	.12
September	August-Atbara August-Khartoum August-Malakal August- Aswan	4	52	.45	.345
October	January-Atbara June-Khartoum September-Aswan	6	60	.30	.18
November	August-Atbara October-Khartoum	7	49	.25	.04

4.3 ANFIS Results

ANFIS is applied with different combinations of variables as inputs and different values of learning parameters. The model with the best performance is chosen for each month. The model with the best performance is the one that gives good results for all the three goodness-of-fit statistics. Based on Table 1, it is obvious that using subtractive clustering method reduces effectively the number of rules compared to grid partition without affecting model performance. For example if we use grid partition with three variables each has three membership functions, we will have 3×3×3= 27 rules. With subtractive clustering the maximum number of rules we use for any month model is 7 rules.

Table 2: ANFIS Goodness-of-fit Statistics using Normalized Discharges.

	RMSE			MAE			R^2		R^2
Data Type Month	Train Data	Test Data	Whole Data	Train Data	Test Data	Whole Data	Train Data	Test Data	Whole Data
July	.054	.073	.056	.037	.061	.039	.938	.935	.899
August	.093	.136	.099	.064	.097	.067	.886	.828	.765
September	.079	.099	.083	.059	.088	.062	.900	.885	.805
October	.074	.082	.075	.048	.073	.051	.942	.905	.884
November	.071	.091	.073	.058	.078	.060	.940	.925	.885

Table 2 shows the goodness-of-fit statistics for river Nile flood forecasts obtained using ANFIS model for each month from July to November. Based on these results, we can see that the models developed using ANFIS for flood prediction at river Nile have high performance degree for all months The three goodness-of-fit statistics reflect such a high performance of ANFIS models. These statistics show that the models are stable, have small errors and in the same time reserve a high and positive correlation between the actual and predicted values for both test and training data. Comparing results obtained for training and testing data, we can say that the developed models do not suffer from overfitting problems and have good generalization ability since the performance of the training data set is near to performance of the testing data set. We observe that August and September have a performance degree smaller than other months; which is expected as flood occurs basically in these months so they have a degree of variability and uncertainty greater than other months. Figures from 3 to 7 show the actual and predicted output for months July-September obtained by ANFIS.

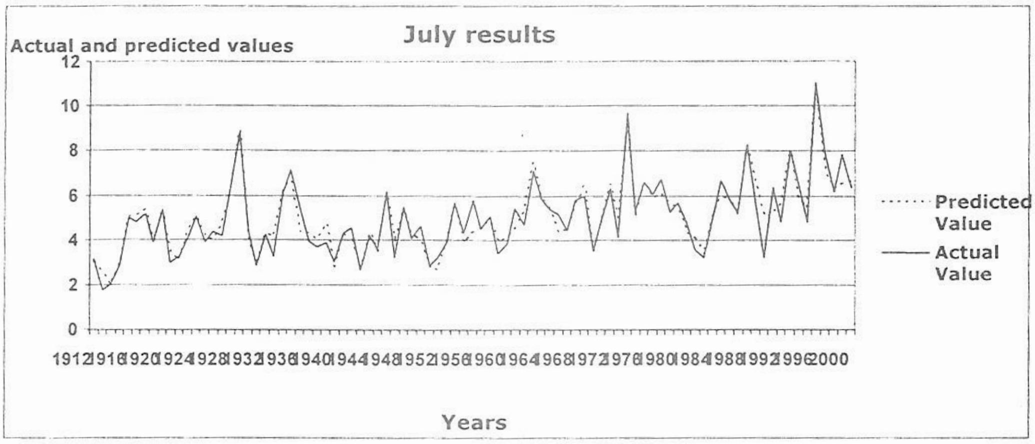

Figure 3: Actual and predicted (using ANFIS) values for July.

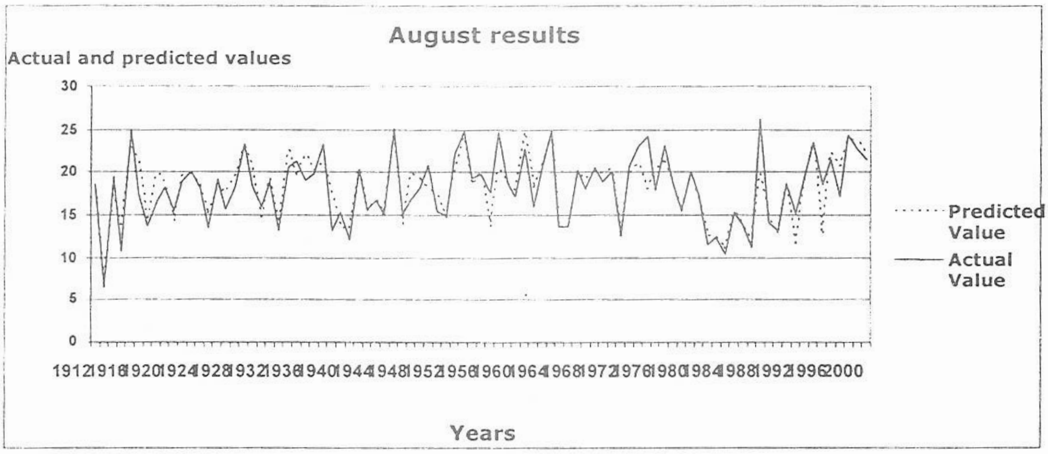

Figure 4: Actual and predicted (using ANFIS) values for August.

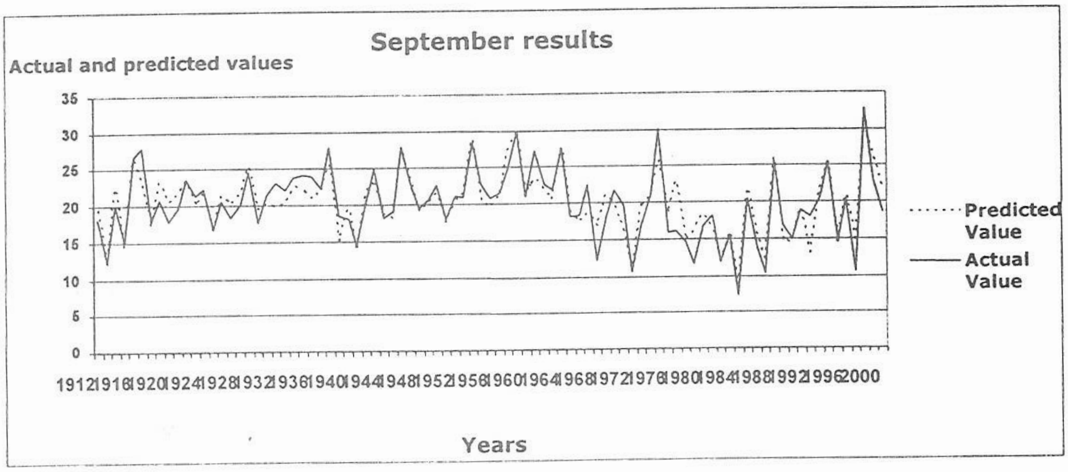

Figure 5: Actual and predicted (using ANFIS) values for September.

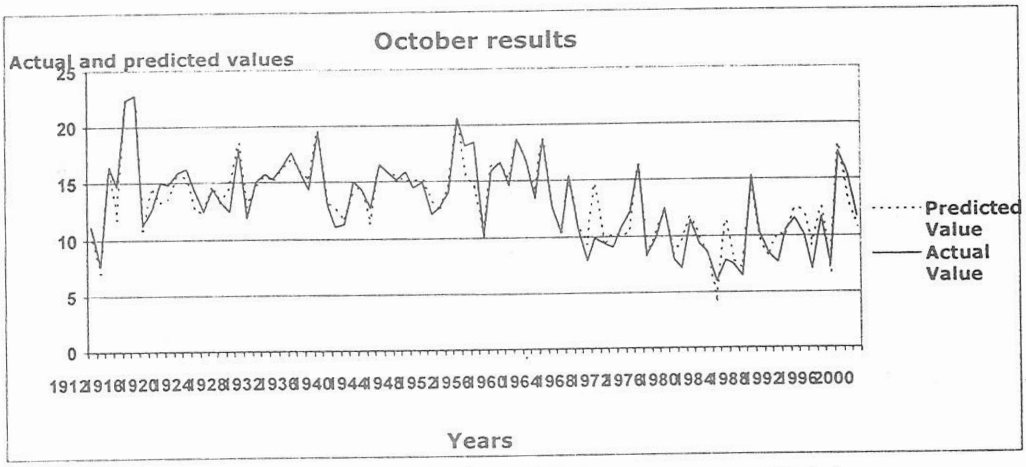

Figure 6: Actual and predicted (using ANFIS) values for October.

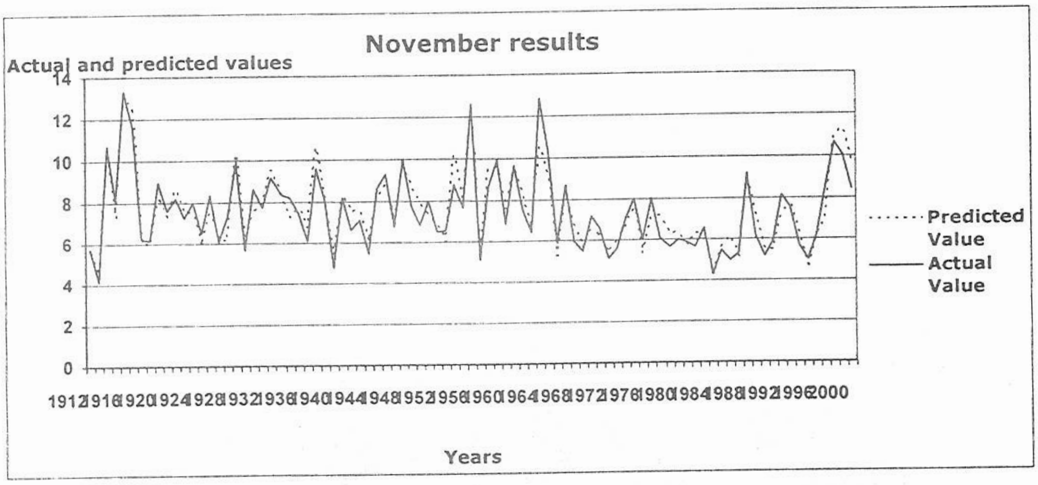

Figure 7: Actual and predicted (using ANFIS) values for November.

Other techniques were used to judge our approach of using ANFIS model to predict flood flow in Egypt at Aswan. These techniques are stepwise regression and neural networks.

In stepwise regression, the previous twelve monthly discharges for each station were used. The size of date set for regression analysis is 88 not 89 pairs because of existence of one lag. The lagged value is assumed to be variable in the previous year since we use the previous twelve variables. Training data set is fixed (80 pairs) and the test data is reduced to 8 instead of 9 pairs. Variables that are selected by regression analysis indicate that river Nile has a memory that varies from one month to the other (see Table 3). Table 4 shows goodness-of-fit statistics obtained using regression analysis technique.

Table 3: Variables used by Regression Analysis.

	Variables Used in Prediction
July	Discharge of Khartoum on June, Malakal on May, Aswan on May
August	Discharge of Atbara on September, Khartoum on June, Khartoum on July, Khartoum on September
September	Discharge of Aswan on July, Aswan on August, Aswan on October
October	Discharge of Atbara on September, Khartoum on March, Khartoum on July, Malakal on January, Aswan on March Aswan on September
November	Discharge of Atbara on Mars, Atbara on December, Khartoum on September, Khartoum on October, Malakal on September, Aswan on April, Aswan on September

Table 4: Regression goodness-of-fit statistics using normalized discharges.

	RMSE			MAE			R^2		R^2
Month / Data Type	Train Data	Test Data	Whole Data	Trained Data	Test Data	Whole Data	Trained Data	Test Data	Whole Data
July	.112	.147	.116	.091	.113	.094	.680	.811	.567
August	.134	.164	.137	.106	.122	.108	.748	.669	.535
September	.106	.201	.118	.082	.175	.092	.815	.599	.608
October	.085	.112	.088	.069	.081	.072	.925	.882	.846
November	.068	.118	.073	.055	.083	.057	.945	.843	.869

one hidden layer as most studies confirm that this is sufficient to approximate any nonlinear function to any degree of accuracy [2]. We make different trials to determine the number of nodes of the hidden layer and we find that 4 nodes for September and 5 nodes for other months in the hidden layer give an acceptable performance. Different learning rates with different number of epochs were used. For each net with a certain learning rate and a certain number of epochs, we run the program more than one time (five times) since the output of neural network depends on the initial weights of the network. We use the same variables as in ANFIS model. The corresponding goodness-of-fit statistics are shown in Table 5.

Table 5: Neural network goodness-of-fit statistics using normalized discharges.

Data Type / Month	RMSE			MAE			R^2		R^2
	Train Data	Test Data	Whole Data	Train Data	Test Data	Whole Data	Train Data	Test Data	Whole Data
July	.081	.110	.082	.061	.081	.063	.889	.827	.778
August	.092	.146	.098	.067	.112	.071	.876	.726	.725
September	.084	.120	.088	.065	.108	.07	.875	.885	.767
October	.091	.108	.093	.067	.078	.069	.914	.90	.826
November	.087	.108	.090	.065	.083	.066	.922	.867	.840

Tables 4 and 5 indicate that stepwise regression gives high performance for October and November and poor performance for the remaining months. Neural networks have higher performance than that obtained from stepwise regression in July, August and September. Neural networks show a reasonable generalization capability however they take a relative long time in training compared to ANFIS. Surprisingly, although July is not in the flood period, these two techniques show bad performance for this month especially when it is compared with performance obtained from ANFIS. Comparison of these two techniques with ANFIS shows that there is a significant difference between the two techniques and ANFIS in favor to ANFIS. This difference is large in case of stepwise regression and relatively small in case of neural networks. This is expected as ANFIS is basically a neural network model that implements fuzzy set techniques.

River flood forecasting is a very difficult problem that needs certain techniques capable of handling nonlinearity, uncertainty and noisy data. A model that combines both neural networks and fuzzy set techniques can help in handling this problem effectively. ANFIS is a model that combines both neural network and fuzzy systems and is known to be effective for approximating nonlinear complex systems and model identification problems. In this paper we use ANFIS for forecasting river Nile flood. Results show that ANFIS is superior to both stepwise regression and neural networks with high performance for all months included in the study. Combining fuzzy systems with neural networks lead to a great improvement in the performance of the model. With ANFIS, the model can be better understood so that the black box problem of neural network is solved. With ANFIS it is easier to generate fuzzy models that achieve a certain degree of accuracy

Some preprocessing data techniques are applied to our data set after naturalizing them. First a feature selection method is applied to select variables with the most predictive power to the underlying process which is done by generating a group of ANFIS models for each month. Feature selection techniques help us in generating powerful models with small number of variables so the curse of dimensionality problem of ANFIS is solved. Second we have normalized our data set to avoid numerical problems. Normalization leads to a great enhancement in the performance of the model compared with its performance in case of using the original data. Models with original data suffer from overfitting problem with high number of parameters. Normalization helps in overcoming these problems effectively by reducing the number of parameters to a high extent. Finally, subtractive clustering is used to initialize ANFIS. This enables ANFIS to distribute the system knowledge between the rules, resulting in a reduced rule base.

For further research a model that combines both neural network and fuzzy set techniques with a feedback connection can be used as flood always depends on previous discharges. Neural networks with fuzzy inputs can be used to increase the ability of neural networks of handling uncertainty. Also it is possible to use neural networks with fuzzy parameter adaptation to reduce time taken in choosing values of neural network's parameters that give best performance which will lead to faster convergence. Additional comparison with the "radial basis function network" can also be suggested to check whether that ANFIS is superior to both of stepwise regression and neural networks. Finally, other efficient neurofuzzy models with higher performance can be applied to this forecasting problem as that given in [21].

Acknowledgment:

The authors are indebted to Professor Nadia Makary for her efforts and advice during the paper preparation.

research, Technical report series, 5/2000, Faculty of information technology, Monash University, Australia.

[2] Atiya, A.F., El-Shoura, S. M., Shaheen, S.I., El-Sherif, M.S. (1999), A comparison between neural network forecasting techniques- Case study: River flow forecasting, *IEEE transactions on neural networks*, vol. 10, No. 2, pp. 402-409.

[3] Babuska, R. (1999), An overview of fuzzy modeling and model-based fuzzy control, *Fuzzy logic control advances in applications*, Eds. Verbruggen, H. B., Babuska, R., World scientific, Singapore.

[4] Babuska, R. (2002), Neuro-fuzzy methods for modeling and identification, *Recent advances in intelligent paradigms and applications*, Eds. Abraham, A., Jain, L.C., Kacprzyk, J., Springer-Verlag, Heidelberg.

[5] Baraldi, A., Blonda, P. (1999), A Survey of fuzzy clustering algorithms for pattern recognition-part I, *IEEE transactions of systems, man, and cybernetics*-Part B, Vol. 29, No. 6, pp. 788-801.

[6] Bowie, M. (2004), *Fuzzy clustering, feature selection, and membership function selection*, Seminar paper, Diuf-department of informatics, University of Fribourg, Switzerland.

[7] Castellano, G., Castiello, C., Fanelli, A. M. (2003), *Discovering prediction rules by a neuro-fuzzy modeling framework*, Proc. of 7th international conference of Knowledge-based intelligent information and engineering systems, Vol. 1, Springer, Oxford, UK, pp. 1242-1248.

[8] Dorrah, H.T (2003), *Analysis and modeling stochastic hydrologic data*, Workshop on analysis of water resources system, Irrigation ministry, Cairo, Egypt.

[9] Haykin, S. (1994), *Neural networks: a comprehensive foundation*, Prentice international, Inc., USA.

[10] Jang, J.-S. R., Sun, C. T., Mizutani, E. (2003), *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*, Prentice Hall, India.

[11] Kneale, P., See, L., Smith, A. (2001), *Towards defining evaluation measures for neural network forecasting models*, Proc. of the 6th international conference of Geocomputation, University of Queen, Slandia.

[12] Li, H., Chen, C. L. P., Haung, H.-P (2000), *Fuzzy neural intelligent systems: Mathematical foundation and applications in engineering*, CRC Press, New York.

[13] Lin, C.-T., Lee, C.S. (1996), *Neural fuzzy systems: A neuro-fuzzy synergism to intelligent*

[15] Nauck, D., Klawonn, F., Kruse, R. (1997), *Foundations of neuro-fuzzy system*, John Wiley& Sons. Inc., USA.

[16] Rajasekaran, S., Pai, G. A. V. (2003), *Neural networks, fuzzy logic, and genetic algorithms: Synthesis and applications*, Prentice Hall, India.

[17] Roskar, J. (2000), *Assessing the water resources potential of the Nile river based on data available at the NFC in Cairo*, Nile Forecast Center NFC, Cairo, Egypt.

[18] See, L., Corne, S.,Dougherty, M., Openshaw, S. (1997), *Some initial experiments with neural network models of flood forecasting on the river Ouse*, Second conference of Geocomputation 97, University of Otago, New Zealand, pp. 15-22.

[19] UNDP (1983), *Master plan for water resources development and use: Multi-lead forecasting for river Nile streamflows,* Technical report 21 UNDP-EGY/81/1031.

[20] Viljamaa, P. (2002), *Fuzzy gain scheduling and tuning of multivariable fuzzy control-Methods of fuzzy computing in control systems*, Ph. D. Thesis, Tampere University of technology, Finland.

[21] M. Figueiredo, R. Ballini, S. soares, M. Andrade, and F. Gomide, Learning Algorithms for a class of Neurofuzzy Network and Application, *IEEE trans. systems, man and cybernetics*, Part C, Vol. 34, No. 3, pp 293-301, August 2004.