Binaya Ku. Panigrahi *et al*, International Journal of Computer Science and Mobile Applications,
National Conference on "The Things Services and Applications of Internet of Things",
Gandhi Institute for Education and Technology (GIET) Baniatangi, 23-24 March 2018, pg. 63-71
**ISSN: 2321-8363**
**Impact Factor: 5.515**

# Flood Forecasting in Brutanga River– A Case Study of Hilly Region, India

**Binaya Ku. Panigrahi[1], T.K Nath[2], M.R. Senapati[3]**

[1]*Ph.D. Scholar, Utkal University, Odisha, panigrahibk1111@gmail.com*
[2]*Professor, IGIT, Sarang, Odisha, nath.tushar@gmail.com*
[3]*Associate Professor, VSSUT, Odisha, manasranjan.senapati@gmail.com*

## Abstract

Flooding in small and medium rivers is seriously threatening the safety of human beings' life and property. The simulation forecasting of the river flood and bank risk in hilly region has gradually become a hotspot. At present, there are few studies on the simulation of hilly perched river, especially in the case of lacking section flow data. And the method of how to determine the position of the levee breach along the river bank is not much enough. Based on the characteristics of the sections in hilly perched river, an attempt is applied in this paper which establishes the correlation between the flow profile computed by HEC-RAS model and the river bank. A hilly perched river in odisha, Brutanga of odisha, india, is taken as the study object, the levee breach positions along the bank are simulated under four different design storm. The results show that the flood control standard of upper reach is high, which can withstand the design storm of 100 years. The current standard of lower reach is low, which is the flooding channel with high frequency. As the standard of current channel between the 2rd and the 11$^{th}$ section is low, levee along that channel of the river bank is considered to be heighten and reinforced. The study results can provide some technical support for flood proofing in hilly region and some reference for the reinforcement of river bank.

***Keywords****: ANN, PSO, MGB-IPH, KNN*

## 1. Introduction

The incorporation of quantitative precipitation forecasting (QPF) in flood warning systems has been acknowledged to play a key role, allowing for an extension of the lead-time of the river flow forecast, which may enable a more timely implementation of flood control [1] (Brath et al., 1988). The QPF integration is particularly needed in small and medium-sized mountainous basins where, given the short response time of the watershed, a precipitation forecast is necessary for an extension of the lead-time of the flood warning. It is widely recognized that obtaining a reliable QPF is not an easy task, rainfall being one of the most difficult elements of the hydrological cycle to forecast (e.g. French et al., 1992), and great uncertainties still affect the performances of both stochastic and deterministic rainfall prediction models.

River flow forecasts are required to provide basic information for reservoir management in a multipurpose water system optimization framework. An accurate prediction of flow rates in tributary streams is crucial to optimize the management of water resources considering extended time horizons. Moreover, runoff prediction is crucial in protection from water shortage and possible flood damages.

The rainfall-runoff, process represents a complex nonlinear problem and there are several approaches to solve it. Traditionally, hydrological simulation modeling systems are classified into three main groups, namely, empirical black box, lumped conceptual, and distributed physically-based models [3, 2].

 

**ISSN: 2321-8363**
**Impact Factor: 5.515**

Flooding leads to numerous hazards, with consequences including risk to human life, disturbance of transport and communication networks, damage to buildings and infrastructure, and the loss of agricultural crops. Therefore, prevention and protection policies are required that aim to reduce the vulnerability of people and public and private property. Many solutions for flood mitigation and prevention have been suggested however, a vast amount of data and knowledge are required about the causes and influencing factors of floods and their resulting damage. Flood forecasting and prediction capabilities evolved slowly during the 1970s and 1980s. However, recent technological advances have had a major impact on forecasting methodologies. For instance, hydrological models use physical detection systems to forecast flood conditions based on predicted and/or measured parameters [2]. River flow models are used as components in actual flood forecasting schemes, where forecasts are required to issue warnings and to permit the evacuation of populations threatened by rising water levels. The basis of such forecasts is invariably observation and/or predictions of rainfall in the upper catchment area and/or river flows at upstream points along main rivers or tributaries. Forecasts about the discharge are obtained in real-time, by using the model to transform the input functions into a corresponding discharge function time [3].

Given the important role of flood forecasting and that so much has been written on the subject, this paper aims to provide comprehensive coverage of the status of the research work carried out by different researchers. Taking a utilitarian viewpoint, we believe that the success of a forecasting model lies in its out-of-sample forecasting power. It is impossible, in practice, to perform tests on all flood forecasting models on a large number of data sets and over many different periods. The contribution of this review is to provide a bird's-eye view of the whole forecasting literature and to provide some recommendations for the practice and future research.

## 2. Literature review

The methods of flood routing are broadly classified a empirical, hydraulic, and hydrological (Fread, 1981). A number of soft computing related techniques were used for flood forecasting in addition to Muskingum method. A brief literature review is presented to provide an overview.

Preliminary concepts and numerous applications of Artificial Neural Networks (ANN) to hydrology are available (ASCE, 2000a,b; Fernando and Jayawardena, 1998). Cheng and Chau (2001), Cheng and Chau (2002) proposed fuzzyiteration methodology and three-person multi-objective conflict decision model respectively for reservoir flood control operation for a case study of Fengman Reservoir, China. Chau et al. (2005) employed the Genetic Algorithm based Artificial Neural Network (ANN-GA) and the Adaptive Network based Fuzzy Inference System (ANFIS), for flood forecasting in a reach of the Yangtze River in China. Similar studies are reported by Cheng et al. (2002, 2008a,b).

Muskingum method is a hydrological flood routing technique (Chow et al., 1988) which was modified by many researchers. In the two parameter Muskingum method, there are number of ways for finding the two parameters, K (travel time) and x (weighing factor for prism and wedge storage of routing reach). These methods were discussed in detail by Singh and McCann (1980) and applied to a set of data to assess their relative efficacy. Gill (1978) proposed segmented curve method, in which least square method was used to find out the parameters of nonlinear form of Muskingum method. Stephenson (1979) demonstrated the way to calculate directly the coefficients of Muskingum method, C0, C1, and C2 using Linear Programming instead of calculating the parameters, K and x.

O'Donnell (1985) considered the lateral flow factor in Muskingum two parameter model of single input single output (si-so) nature, which was converted into a three parameter model. The parameters are K, x, a (a shows

 

Binaya Ku. Panigrahi *et al*, International Journal of Computer Science and Mobile Applications,
National Conference on "The Things Services and Applications of Internet of Things",
Gandhi Institute for Education and Technology (GIET) Baniatangi, 23-24 March 2018, pg. 63-71
**ISSN: 2321-8363**
**Impact Factor: 5.515**

the fraction of lateral flow in comparison with inflow to the reach). The least square technique is used to find out these parameters in the routing reach automatically. Khan (1993) extended the si-so flood routing model to include lateral flow to form a multi input single output (mi-so) model with lateral flow.

Tung (1985) developed state variable modeling technique for solving the nonlinear form of Muskingum method. The parameters of the model were found out by four methods of curve fitting. Yoon and Padmanabhan (1993) developed software, MUPERS, where both linear and nonlinear relationships were dealt with. Kshirsagar et al. (1995) found parameters by a constrained, nonlinear (successive quadratic) programming. In this work, the Muskingum equation was used for routing the upstream hydrograph and the intermediate un gauged lateral inflow. The lateral inflow was calculated by an impulse response function approach. Mohan (1997) used genetic algorithm for parameter estimation of nonlinear Muskingum method and compared its performance with the approach by Yoon and Padmanabhan (1993).

Samani and Jebelifard (2003) applied multi linear Muskingum method for hydrologic routing through circular conduits. Das (2004) developed a methodology for parameter estimation for the Muskingum model of stream flow routing. Al-Humond and Esen (2006) presented two approximate methods for estimating Muskingum flood routing parameters. Geem (2006) introduced the Broydene Fletchere Goldfarbe Shanno (BFGS) technique, which searches the solution area based on gradients for estimation of Muskingum parameters.

Choudhury (2007) proposed a multiple inflows Muskingum model. This model appropriately extended the Muskingum philosophy to multiple inflows routing, expressed in a single in flow single out flow form. The model performance is compared with the nonlinear kinematic wave model. He applied the model to the flood events in Narmada Basin, India. Das (2007) developed a chance constrained optimization based model, for Muskingum model parameter estimation. Das (2009) developed a methodology for Muskingum model's parameter estimation for reverse stream flow routing for which a fresh calibration was found necessary. Chu (2009) applied Fuzzy Inference System (FIS) and Muskingum model in flood routing where rules of FIS were incorporated with the Muskingum formula.

## 3. Comparing different Rain fall and Flood forecasting Techniques:

### 3.1. Artificial Neural Network:

An alternative approach to flow forecasting has been developed in the recent years, which is based on the ANN [3]. Recent studies have reported that ANN may offer a promising alternative for the hydrological forecasting of stream flow [7]. The ANN is a computer program that is designed to model the human brain and its ability to learn tasks [4]. An ANN differs to other forms of computer intelligence in that it is not rule based, as in an expert system. An ANN is trained to recognize and generalize the relationship between a set of inputs and outputs. Early artificial neural networks were inspired by perceptions of how the human brain operates. In the recent years, ANN technological developments have made it more of an applied mathematical technique with some similarities to the human brain. ANNs retain two characteristics of the brain as primary features: the ability to (1) 'learn' and (2) generalize from limited information [5]. Both biological and artificial neural networks employ massive, interconnected simple processing elements, or neurons. The knowledge stored as the strength of the interconnecting weights (a numeric parameter) in ANNs is modified through a process called learning, using a learning algorithm. This algorithmic function, in conjunction with a learning rule, (i.e., back-propagation) is used to modify the weights in the network in an orderly fashion. Unlike most computer applications, an ANN is not ''programmed,'' rather it is ''taught'' to give an acceptable answer to a particular problem. Input and output values are sent to the ANN, initial

 

**ISSN: 2321-8363**
**Impact Factor: 5.515**

weights to the connections in the architecture of the ANN are assigned, and the ANN repeatedly adjusts these interconnecting weights until it successfully produces output values that match the original values. This weighted matrix of interconnections allows the neural network to learn and remember [10]. When using an ANN to solve a problem, the first step is to train the ANN to ''learn'' the relationship between the input and outputs. This action is accomplished by presenting the network with examples of known inputs and outputs, in conjunction with a learning rule. The ANN maps the relationship between the inputs and outputs, and then modifies its internal functions to determine the best relationship that is be represented by the ANN.

The inner workings and processing of an ANN are often thought of as a ''black box'' with inputs and outputs. One use-ful analogy that helps to understand the mechanism occurring inside the black box is to consider the neural network as a super-form of multiple regressions. Like linear regression, which finds the relationship that {y} = f{x}, the neural network finds some function f{x} when trained. However, the neural network is not limited to linear functions. It finds its own best function to the best of its ability, given the complexity used in the network, and without the constraint of linearity (Hewitson and Crane [5]). The most common type of artificial neural network consists of three groups, or layers, of units: (1) a layer of ''input'' units are connected to (2) a layer of ''hidden'' units, which are connected to (3) a layer of ''output'' units (Fig. 1)The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input units and the hidden units. The behavior of the output units depends on the activity of the hidden units and the weights between the hidden units and output units [12].

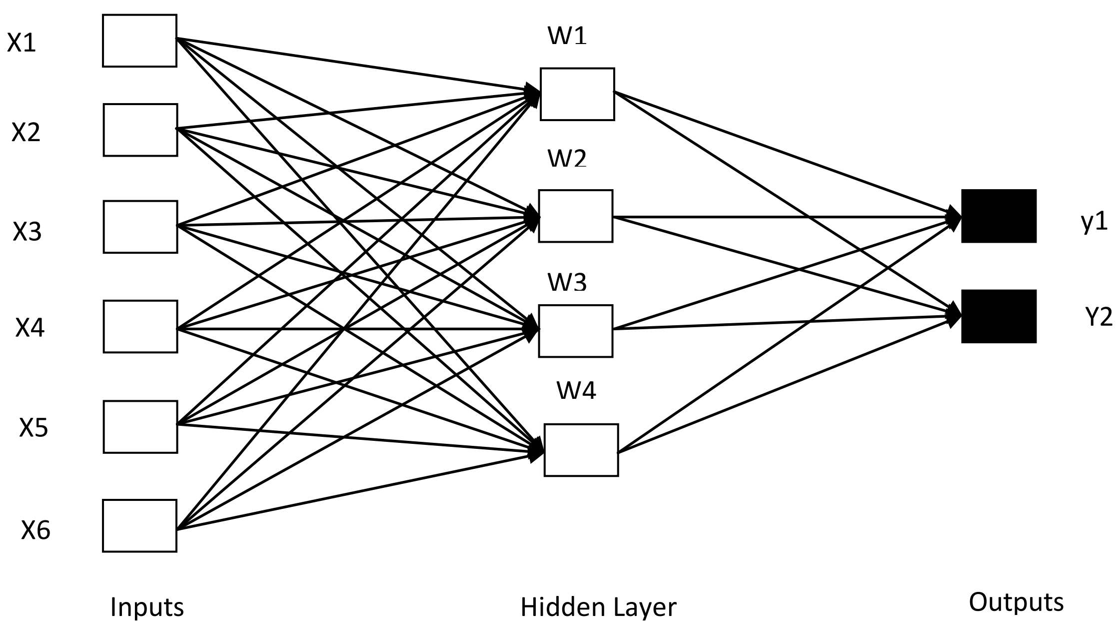


**Figure 1:** Simple feed forward network. http://dx.doi.org/10.1016/j.aej.2014.06.010

 

Binaya Ku. Panigrahi *et al*, International Journal of Computer Science and Mobile Applications,
National Conference on "The Things Services and Applications of Internet of Things",
Gandhi Institute for Education and Technology (GIET) Baniatangi, 23-24 March 2018, pg. 63-71
**ISSN: 2321-8363**
**Impact Factor: 5.515**

### 3.2 PSO algorithm

The principle of PSO algorithm is founded on the assumption that potential solutions will be flown through hyperspace with acceleration towards more optimum solutions. It is a populated search method for optimization of non-linear functions resembling the movement of organisms in a bird flock or fish school. Candidate solutions to the problem are termed particles or individuals. Instead of employing genetic operators, the evolution of generations of a population of these individuals in such a system is by cooperation and competition among the individuals themselves. In essence, each particle adjusts its flying based on the flying experiences of both itself and its companions. During the process, it keeps track of its coordinates in hyperspace which are associated with its previous best fitness solution, and also of its counterpart corresponding to the overall best value acquired thus far by any other particle in the population. In the algorithm, vectors are taken as representation of particles since most optimization problems are convenient for such variable presentations. The population is responding to the quality factors of the previous best individual values and the previous best group values. The allocation of responses between the individual and group values ensures a diversity of response. Its major advantages are the relatively simple and computationally inexpensive coding and its adaptability corresponding to the change of the best group value. The stochastic PSO algorithm has been found to be able to find the global optimum with a large probability and high convergence rate (Clerc and Kennedy, 2002).

Hence, it is adopted to train the multi-layer perceptrons, within which matrices learning problems are dealt with. Adaptation to network training A three-layered perceptron is chosen for this application case. Here, W[1] and W[2] represent the connection weight matrix between the input layer and the hidden layer, and that between the hidden layer and the output layer, respectively. When a PSO is employed to train the multi-layer perceptrons, the ith particle is denoted by

$$Wi = \{ Wi [1], Wi [2] \} \tag{1}$$

The position representing the previous best fitness value of any particle is recorded and denoted by

$$pi = \{ pi [1], pi [2] \} \tag{2}$$

If, among all the particles in the current population, the index of the best particle is represented by the symbol b, then the best matrix is denoted by

$$pi = \{ pb [1], pb [2] \} \tag{3}$$

The velocity of particle i is denoted by

$$Vi = \{ Vi [1], Vi [2] \} \tag{4}$$

If m and n represent the index of matrix row and column, respectively, the manipulation of the particles are as follows

$$Vi``[j] (m, n) = Vi[j] (m, n) + \{ r\alpha [Pi [j] (m , n) - Wi[j] (m, n) ] + s\beta[ Pb [j] (m , n) - Wi[j] (m, n) ] \} / t \tag{5}$$

 

Binaya Ku. Panigrahi *et al*, International Journal of Computer Science and Mobile Applications,
National Conference on "The Things Services and Applications of Internet of Things",
Gandhi Institute for Education and Technology (GIET) Baniatangi, 23-24 March 2018, pg. 63-71
**ISSN: 2321-8363**
**Impact Factor: 5.515**

And

Wi``[j] = Wi[j] + Vi[j] t

where j = 1, 2; m = 1,. . . ,Mj; n = 1,. . . ,Nj; Mj and Nj are the row and column sizes of the matrices W, P, and V; r and s are positive constants; α and β are random numbers in the range from 0 to 1; t is the time step between observations and is often taken as unity; V`` and W`` represent the new values. Eq. (5) is employed to compute the new velocity of the particle based on its previous velocity and the distances of its current position from the best experiences both in its own and as a group. In the context of the social behavior, the cognition part, i.e., the second element on the right hand side of Eq. (5), represents the private thinking of the particle itself whilst the social part, i.e., the third element on the right hand side of Eq. (5), denotes the collaboration among the particles as a group. Eq. (6) then determines the new position according to the new velocity.

The fitness of the ith particle is expressed in term of an output mean squared error of the neural networks as follows

$$f(Wi) = \frac{1}{s}\sum_{k=1}^{s}\left[\sum_{t=1}^{0}\left\{t_{kl} - p_{kl}\,(W_i)\right\}^2\right] \qquad (6)$$

where f is the fitness value, tkl is the target output; pkl is the predicted output based on Wi; S is the number of training set samples; and, O is the number of output neurons.

### 3.3 The MGB-IPH hydrological model

Many hydrological models can be used to make stream flow forecasts based on predicted rainfall, and the comparative study developed in Brazil by ONS also included lumped rainfall–runoff models, more complex distributed hydrological models, and black-box models based on neural networks. The question whether a distributed rainfall–runoff model performs better than simpler models has been posed repeatedly in the past. It has been argued that distributed models would perform better where distributed input data were available, such as rainfall estimated by radar. But a recent study – the Distributed Model Inter comparison Project – showed that lumped models performed comparatively well even using radar rainfall data, although in one basin with elongated shape (Blue River), distributed models outperformed lumped models (Reed et al., 2004). Distributed models also appear to perform better when uncertainties in input data and parameter values are considered (Carpenterand Georgakakos, 2006). It can also be argued that distributed or semi-distributed models should be used in large basins where spatial variability in rainfall and runoff generation processes may play a larger role, and the results presented in this paper were all obtained using the distributed large-scale hydrological model MGH-IPH (Collischonn et al., 2007; Collischonn and Tucci, 2001). This is a large-scale distributed hydrological model developed for use in large South American basins, where densities of hydrological instrument networks are relatively low and records are commonly short. Using the classification proposed by Beven (2001), the model can be classified as a hydrological response unit model. It uses input data derived from Geographical Information Systems giving information on basin characteristics such as land use, topography, vegetation cover and soil types, which guide the calibration of parameter values. The MGB-IPH model was developed from the LARSIM (Bremicker, 1998) and VIC (Liang et al., 1994; Nijssem et al., 1997) models, with some changes in the evapotranspiration, percolation and stream flow propagation modules. It has modules for calculating the soil water budget; evapotranspiration; flow propagation within a cell, and flow routing through the drainage network. The drainage basin is divided into elements of area –

**ISSN: 2321-8363**
**Impact Factor: 5.515**

normally on a Medium-range reservoir inflow predictions based on quantitative precipitation forecasts 113 square grid of 10 · 10 km – connected by channels, with vegetation and land use within each element categorized into one or more classes, the number of vegetation and land-use types being at the choice of the user. The Grouped Response Unit (GRU) (Kouwen et al., 1993) approach is used for hydrological classification of all areas with a similar combination of soil and land cover without consideration of their exact locality within the grid (or cell). A cell contains a limited number of distinct GRUs. Soil water budget is computed for each GRU, and runoff generated from the different GRUs in the cell is then summed and routed through the river network. This approach has been used in other large-scale hydrological models, such as VIC (Wood et al., 1992; Liang et al., 1994; Nijssem et al., 1997) and WATFLOOD (Kouwen and Mousavi, 2002; Soulis et al., 2004).

The soil water balance is computed independently for each GRU of each cell, using components describing canopy interception, evapotranspiration, infiltration, surface runoff, sub-surface flow, base flow and soil water storage. Rainfall values are interpolated spatially and at each time step to give an estimate at the center of each grid cell using inverse-distance-squared interpolation. Flow generated within each cell is routed to the stream network using three linear reservoirs (baseflow, sub-surface flow and surface flow). Stream flow is propagated through the river network using the Muskingum–Cunge method. A more comprehensive description of the model, including results from aproxy-basin test, is given by Collischonn et al. (2007) and further applications are presented by Allasia et al. (2006), Collischonn et al. (2005) and Tucci et al. (2003).

### 3.4. The KNN Method

The K-nearest-neighbor method has its origins as a non-parametric statistical pattern recognition procedure, aiming at distinguishing between different patterns according to chosen criteria. Among the various non-parametric techniques, in the sense that no theoretical or analytical relation is known or assumed between the inputs and the outputs, it is the most intuitive, but nevertheless possesses powerful statistical properties. Yakowitz (1987) and Karlsson and Yakowitz (1987a,b) did considerable work in extending the K-NN method to time-series and forecasting problems, obtaining satisfactory results and constructing a robust theoretical base for the K-NN method. The intuitiveness of the approach and the powerful theoretical basis have made the method attractive to forecasters, especially in the hydrologic field, where the method found successful applications (Karlsson and Yakowitz, 1987a,b; Galeati, 1990; Kember and Flower, 1993; Todini, 1999).

The prediction of a time series is based on a local approximation, making use of only the nearby observations. For each forecast instant t, let $X^{-d}(t)$ = (X t , …… X t-d+1 ) . be a feature vector of past records. A feature vector is a vector that summarizes the whole past history in a smaller-dimension vector of observations supposed to contain most of the information relevant to the forecast. The method assumes that the probability distribution of the random variable conditioned on the entire past .xt+1 / xt ; xt-1;….) ; is the same as that of the random variable conditioned on only the d past observations ( xt+1/ $X^{-d}(t)$ ): It was proved that, even if $X^{-d}(t)$ does not satisfy the above “history summarization” properties, the K-NN forecaster will be asymptotically optimal among all the forecasters defined on the feature vector $X^{-d}(t)$: That is, under fairly general circumstances, convergence to the optimal forecaster is assured as the historical data set increases (Karlsson and Yakowitz, 1987b). Let us indicate the expectation of the next value as $\widehat{x_{t+1}}$, conditioned on the current feature vector $X^{-d}(t)$:; that is, $\widehat{x_{t+1}} = E\,[x_{t+1} / ]$ . To estimate $X^{-d}(t)$ ; the K-NN method imposes a metric, denoted by ‖ . ‖, on the feature vector $X^{-d}(t)$. to find the set of K past nearest neighbours of $X^{-d}(t)$.; i.e. the K ddimensional vectors of past observations: $X^{-d}(t)$.; J = 1,…;

**ISSN: 2321-8363**
**Impact Factor: 5.515**

K; which minimise $\| X^{-d}(t) - X^{-d}(t_j) \|$The most intuitive and widely used metric to identify neighbours is the Euclidean norm, which, for a d-dimensional vector $Z^{-d}$ = ( Z1, Z2 ,…. Zn) in $\| Z^{-d} \| = (\sum_{i=1}^{d} {z_i}^2)^{1/2}$ (16)

The forecast is then obtained by averaging the temporal evolution of the nearest neighbors, assumed to be similar to the evolution of the current situation, that is,

$$\hat{x}_{t+1} = \frac{1}{k} \sum_{j=1}^{k} x_{t_{j+1}} \quad (17)$$

The generalisation to higher lead-times L is straightforward:

$$\hat{x}_{t+L} = \frac{1}{k} \sum_{j=1}^{k} x_{t_{j+L}} \quad (18)$$

### 4. Objective of the research

In this research, water flow data of Brutanga and Bhargovi River will be taken for classification and subsequently predicting the water flow in both the rivers in a particular interval using well known data mining technique like the artificial neural network (ANN), Particle Swarm Optimization, Genetic Algorithm etc. The results obtained from various techniques will then be analyzed to find out the technique(s) giving the best result. The main objective is to warn the people residing downstream well in advance, if there is any possibility of flash flood thereby saving precious lives and valuables.

### 5. Conclusion

Thus, in our case, the K-NN algorithm looks through all consecutive d-dimensional vectors in the entire historical rainfall depths database and locates K of these d-ples, which are closest to the vector of d most recent rainfalls. The prediction of the next rainfall is then taken to be the average of the rainfall subsequent to these K historical nearest neighbors. It may be noticed that the K-NN approach does not require the selection of a class of models and the estimation of the model parameters, so that the identification of a specific form of the input/output relationship is not needed.

# References

1. R. Baratti,B. Cannas,A. Fanni,M. Pintus,G.M. Sechi,River flow forecast for reservoir management, through neural networks, www.elsevier.com/locate/neucom doi:10.1016/S0925-2312(03)00387-4
2. Sulafa Hag Elsafi, Artificial Neural Networks (ANNs) for flood forecasting at Dongola Station in the river Nile,Sudan, Sudan,doi.org/10.1016/j.aej.2014.06.010
3. Nile Basin Capacity Building Network (NBCBN). Flood and Drought, Forecasting and Early Warning Program, 2005.
4. Moore, R.J., Jones, D.A., Black, K.B., Austin, R.M.,Carrington, D.S., Tinnion, M.,Akhondi, A., 1994. RFFS and HYRAD: Integrated System for Rainfall and River Flow Forecasting in Real-Time and their Application in Yorkshire.BHS Ossasional paper No. 4, 12.
5. G.E.P Box, G.M. Jenkins, Time Series Analysis: Forecasting control, Holden –Day, Oakland, California, 1976.
6. S.J. Yakowitz, Markov flow models and the flood warning problem, Water Res Res 21 (1985) 81–88.

 

**ISSN: 2321-8363**
**Impact Factor: 5.515**
7. D. Nagesh Kumar , Falguni Baliarsingh, K. Srinivasa Raju , Extended Muskingum method for flood routing, doi:10.1016/j.jher.2010.08.003
8. Govindoraju, R.S., Rao, A.R., Artificial neural networks in hydrology. Netherlands, 2000.
9. Ozgur. Kisi, A combined generalized regression neural network wavelet model for monthly stream flow prediction, KSCE J.Civil Eng. 15 (8) (2011) 1469–1479.
10. Haykin, S., Neural networks. http://www.cul.salk.edu/.tewon/ICA/teaching-KAIST/references.htmc/1994.
11. Anderson, Dave, McNeil, George. Artificial neural networks technology. Data and analysis Centre for Software, Rome,August 1992 <http://www.dtic.mil>.
12. G.E.P Box, G.M. Jenkins, Time Series Analysis: Forecasting control, Holden – Day, Oakland, California, 1976.
13. B.C Hewiston, Crane, Precipitation Controls in SouthernMexico, in Neural Nets, Kluwer Academic Publisher, 1994.
14. Santosh. Patil, Sharda. Patil and Shriniwas. Valunjkar , Study of Different Rainfall-Runoff Forecasting Algorithms for Better Water Consumption, International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2012) Penang, Malaysia
15. MariuszZiejewski ,Goettler,H.J., Comparative analysis of the exhaust emissions for vegetable oil based alternative fuels Society of Automotive Engineers (1992) Paper No:920195.
16. Gaurav Srivastava , Sudhindra N. Panda , Pratap Mondal , Junguo Liu , Forecasting of rainfall using ocean-atmospheric indices with a fuzzy neural technique, doi:10.1016/j.jhydrol.2010.10.025.
17. K. W Chau, Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River, doi:10.1016/j.jhydrol.2006.02.025.
18. Fangqiong Luo and Jiansheng Wu2010 ” Rainfall Forecasting Using Projection Pursuit Regression and Neural Networks”IEEE2010 Third International Joint Conference on Computational Science and Optimization pp 488 to 491.
19. E. Toth*, A. Brath, A. Montanari , Comparison of short-term rainfall prediction models for real-time flood forecasting.

 