International Journal of Advanced Technology & Engineering Research (IJATER)
National Conference on Recent Trends in Science, Technology & Management (NCRTSTM-2018)

RAINFALL AND FLOOD SITUATION FORECASTING : A REVIEW

Binaya Ku. Panigrahi1, T.K Nath2, M.R. Senapati3
1Ph.D. Scholar, Utkal University,Odisha ,panigrahibk1111@gmail.com
2Professor, IGIT, Sarang, Odisha, nath.tushar@gmail.com
3Associate Professor, VSSUT, Odisha, manasranjan.senapati@gmail.com

Abstract

Forecasts of water inflow into major reservoirs of different rivers are needed for the operational planning over periods ranging from a few hours to several months ahead. Medium-range forecasts of the order of a few days to two weeks have usually been obtained by simple ARMA-type models, which do not utilize information on observed or forecast precipitation, nor stream flow observations from upstream gauging stations. Recently, several different hydrological models have been tested to assess the potential improvements in forecasts that could be obtained by using observed and forecast precipitation as additional inputs. In this paper we have carried out a review of different techniques used for forecasting the water flow into various rivers and fore casting the flood situation.

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].

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.

Fuzzy-Ranking Algorithm (FRA)

Identification of significant input variables is one of the most important steps in the development of a prediction model. To capture the linear or nonlinear relationship between the model inputs and outputs, two-stage Fuzzy-Ranking Algorithm proposed by Lin et al. (1998) was used in this study. The fuzzy ranking process

begins with the construction of fuzzy curves and surfaces for each input variable. Let for an output y there are n possible input variables, x1, x2, . . . , xn. Each variable consists of M data points.

The single performance index for fuzzy curve (PCi) is given as

$$\text{PC i} = \frac{{P_y}^{i}{}_{c}}{1+{pv^{i}}_{c}} \qquad (1)$$

Where ${P_y}^{i}{}_{c}$ and ${pv^{i}}_{c}$ are the first stage and second stage performance indices for fuzzy curve respectively.

For fuzzy surface the single performance index (PSi,j) is defined as

$$\text{PS I,j} = \frac{{P_y}^{i,j}{}_{s}}{1+{pv^{i,j}}_{s}} \qquad (2)$$

where ${P_y}^{i,j}{}_{s}$ and ${pv^{i,j}}_{s}$ are the first and second stage performance indices for fuzzy surface, respectively.

Once the fuzzy curves and surfaces have been generated, they are analyzed in order to determine which input variables are best able to predict the output variables. The FRA uses the performance index to rank the inputs. The performance index is a method that involves checking the mean square error between the fuzzy curve for the variable xi and the output variable y. A small value of this performance index indicates that the variable is related to the output. A similar approach may also be taken for the fuzzy surfaces, which can also give information about whether the two variables are correlated. The FRA then normalizes the performance indices for the fuzzy curves and surfaces. This is carried out by computation of fuzzy curves and surfaces for a random variable generated by computer program. The performance index for the fuzzy curve of xi is divided by the performance index of the simulated random variable in order to normalize it. Fig. 2 shows the flowchart of the FRA used in this study. The FRA applied in this study can be summarized in the following steps:

1. Add a test random variable R to the input set. Designate it as xR.

2. Choose a, 0 < a 6 1 (typically 0:99 < a 6 1).

3. Generate fuzzy curve list and sort by their fuzzy curve performance index (PCi). The variable xj with smallest valve of PCi is regarded as the most important input variable. Eliminate all variable other than the known random variable xR, where PCi=PCR > α from additional consideration since they are apparently only randomly related to the output.

4. Use the most important variable from the last step, say xj with remaining xk, k ≠ j, to generate fuzzy surface (si,j). The input variable xm with the smallest fuzzy surface index (Psj,m) is regarded as the next most important. Eliminate all variable other than xR where Psj;k=Psj;R > alpha or Psj;k=Pcj > α from additional consideration. Xm is selected for next significant variable.

5. Repeat step 4 until no more variables can be eliminated.

FRA was applied between August rainfall and three sets of inputs:

(a) Model (a): Climatic indices (SOI and PDOI) with lag 1–12 months.

(b) Model (b): SSTa with lag 1–12 months.

(c) Model (c): SOI, PDOI and SSTa with lag 1–12 months.

3.4. Data division approach

Three different approaches were followed for the division of data in training, testing, and validation sets for neural network.

1. Random approach.

2. Self-organized map (SOM) approach.

3. Proposed fuzzy c-mean clustering approach.

A new data division approach is proposed in this paper. The proposed data division approach is based on fuzzy c-means clustering.

The fuzzy c-means clustering algorithm is based on the minimization of an objective function called c-means functional. It is defined by Dunn (1973) as:

$$J(X;U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_{ik})^{m} \parallel Xk - vi \parallel A\ 2 \quad (3)$$

where V = [v1; v2; v3; . . . ; vc]; vi ϵ Rn is a vector of cluster prototypes (centers), which have to be determined, and DikA2 =‖ Xk - vi‖ A2 = (Xk – Vi)TA (Xk – Vi) of X k from Vi allows μik in [0,1]. is a squared innerproduct distance norm. Statistically, (9) can be seen as a measure of the total variance of xk from vi.

A fuzzy partition can be seen as a generalization of a hard partition, as it allows μik attaining real values in [0, 1]. A N x c matrix U= [μik] represents the fuzzy partitions; its conditions are given by:

μik = [0,1] , 1 ≤ i ≤ N, 1≤ k ≤ c

$$0 < \sum_{i=1}^{N} \mu_{ik} < N , 1 \leq k \leq c$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \left({D_{ikA}}/{D_{ikA}} \right)^{\frac{2}{(m-1)}}} \qquad (4)$$

where 1 ≤ i ≤ N, 1≤ k ≤ c and

$$v_i = \frac{\sum_{k=1}^{N} \mu_{i,k}{}^{m} \times k}{\sum_{k=1}^{N} \mu_{i,k}{}^{m}} \ , \ 1 \leq k \leq c \qquad (5)$$

where vi is the cluster center. Once the clusters are formed the total information content is computed to identify the optimal numbers of clusters.

Let Cji be ith cluster at jth level (Lj). We measure the Net Information Gain (NIG) during the evolution from Li to Li+1. The gain or loss of information on cluster j from Li to Li+1 is given by: gi = di × Mi (6)

where di is the direction (increase or decrease); and Mi is the magnitude of change in information. If the offspring of cluster j overlap, information is deemed to have been lost and di = -1. In contrast, if offspring are clearly separated without overlap, information is deemed to have been gained and di = -1. The magnitude of information is measured using information theory.

$$\text{Mj} = - \textstyle\sum_{k} p_k \ln p_k \qquad (7)$$

where k is the number of offspring of cluster j and Pk is the fraction of elements migrated from cluster j to kth offspring. Total information content (Ii) is

$$\text{Ii} = \textstyle\sum_{L_1}^{L_i} \sum_{j=1}^{i} g_i \qquad (8)$$

The level with largest information content is considered to be optimal and the number of cluster corresponding to that level is optimal. For optimal number of clusters the data set is divided into three subsets (training, testing, and validation subsets). For each cluster and each membership value interval (interval of 0.0– 0.1; 0.1–0.2; . . . ; 0.9–1) two data points (samples) are chosen, one is assign to testing set and the other one is assign to validation set. All the remaining samples are assigned to training set. If there are only two samples then one will be assigned to testing and the other one to training. In case there is only one sample then it has to be assigned to training set. This data division approach can be summarized in following steps:

1. Initial number of cluster is equal to 1.

2. The available data set are clustered using fuzzy c-mean clustering and the information content of the whole data set is computed.

3. Increase the number of clusters by 1 and repeat the step 2 until number of clusters reaches 50% of available data.

4. The level with maximum information content considered as being optimal and number of clusters corresponding to that level is optimal number of clusters.

5. For optimal number of clusters the data set is divided into three subsets (training, testing, and validation subsets). For each cluster and each membership value interval (interval of 0.0–0.1; 0.1–0.2; . . . ; 0.9–1) two data points (samples) are chosen, one is assigned to testing set and the other one is assigned to validation set. All the remaining samples are assigned to training set. If there are only two samples then one will be assigned to testing and the other one to training. In case there is only one sample then it has to be assigned to training set.

ARMA models

Most of the time-series techniques traditionally used for modeling water resources series fall within

the framework of the ARMA class of linear stochastic processes. They are usually denoted as ARMA (p,q) models, where p and q are the auto-regressive and moving-average orders, respectively (Box and Jenkins, 1976; Brockwell and Davis, 1987; Bras and Rodriguez-Iturbe, 1994). They describe each observation of the time series as a weighted sum of p previous data, and the current as well as q previous values of a white noise process.

The mean of the time series. Parameter estimation for ARMA models can be performed in several ways. We applied here an approximation in the spectral domain of the Gaussian maximum likelihood function, which was first proposed by Whittle (1953) for short-memory models.

1ARMA model application

The application of low-order ARMA processes to model short-term precipitation values is considered here, following the modeling framework proposed by Brath et al. (1988) and Burlando et al. (1993).

The application of ARMA models requires the data to be stationary and this is often not the case for hourly rainfall observations, whose statistical properties may vary with the season. Nonetheless, the limited number of rainfall events in the observation period prevented us, in the split-sample calibration, from grouping the events in monthly periods, as it is usually done in hydrology to circumvent non-stationary. In the adaptive calibration application, non-stationary is accounted for by allowing the model parameters to vary with time since the calibration is performed solely on the progress of the current event. We preferred not to perform any preliminary transformation of the data in order to make them as close to Gaussian as possible. In fact, Gaussian data are not required for the forecast application of ARMA models, since they provide the best linear prediction even in the non-Gaussian case (Brockwell and Davis, 1987).

The selection of the model orders, p and q, was driven by some results available in literature. Obeysekera et al. (1987) determined an equivalence between the correlation structure of an ARMA(1,1) model and some point process models, like the Poisson rectangular pulses and the Neyman–Scott white noise models (see Rodriguez-Iturbe et al.,1984). On the other hand, the Neyman–Scott rectangular pulses model, which has proved to represent the stochastic structure of rainfall better (Rodriguez-Iturbe et al., 1987), has a correlation structure equivalent to that of an ARMA(2,2) process. In the adaptive calibration, the parameters are estimated in correspondence with each forecast instant, on the basis of the last values measured in real-time. The number of past observations to be used for each calibration was chosen on the basis of the results of a previous study (Brath et al. 1998). The estimation of the parameters was performed there with a number w of observations xt immediately preceding each forecast instant, with w varying from 7 to 100, aiming at identifying the value of w that provides the best forecasting performances. The results showed that for increasing w, the efficiency of the forecast improved moderately for short lead times (1–3 h), but a longer set of past data (more than 3 days of previous hourly observations) provided a much better performance for lead-times longer than 4 h. Thus, we set the moving window of past rainfall observations to be used in each adaptive calibration equal to the 100 last measured hourly observations (that is, w = 100)

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,…; 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 $\| Z^{-d} \| = (\sum_{i=1}^{d} z_i{}^2)^{1/2}$ (9)

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,

$$\hat{x}_{t+1} = \frac{1}{k}\sum_{j=1}^{k} x_{t_{j+1}} \qquad (10)$$

The generalisation to higher lead-times L is straightforward:

$$\hat{x}_{t+L} = \frac{1}{k}\sum_{j=1}^{k} x_{t_{j+L}} \qquad (11)$$

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.

[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.