# Implementing Neural Network Approaches for Flood Forecasting: Case Study of Godavari Basin, India

Puneet Misra[1], Shobhit Shukla[2]

[1]Assistant Professor (Stage-III), Department of Computer Science, University of Lucknow, Lucknow, U.P., India.

[2]Research Scholar, Department of Computer Science, University of Lucknow, Lucknow, U.P., India.

*Abstract*—Floods are very dangerous force of nature that have the capability to cause extensive damage to infrastructure and result in great loss of human lives. They also have long lasting effects resulting in waterborne diseases, scarcity of clean drinking water and food and adverse effect to local economy. Floods are chiefly caused by increased river flow due to heavy rains, melting of snow etc. leading to river overflow and breaking its banks thus flooding the area. In this study, the potential of two data driven techniques namely, Artificial Neural Networks (ANN) and Support Vector Machine (SVM) were used for forecasting floods by predicting river flow in lower Godavari river sub-basin of eastern India. The techniques were applied on various models constructed from combinations of various antecedent river flow values from two gauging stations and the results were compared for the best fit models of each technique. To get more accurate assessment of results of the models, three standard statistical quantitative performance assessment parameters, the Mean Squared Error (MSE), the coefficient of correlation (R) and the Nash-Sutcliffe coefficient (NS) were used to analyze the performances of the models developed. The results obtained demonstrate that the performance of all models was satisfactory. A complete comparison of the overall performance indices demonstrated that the SVM model performed better than ANN models in flood prediction.

*Index Terms*—Artificial Neural Networks (ANN), Support Vector Machine (SVM), Mean Squared Error (MSE), the coefficient of correlation (R), Nash-Sutcliffe coefficient (NS)

## I. INTRODUCTION

Floods can result in devastating consequences and can destroy economy, environment often resulting in great loss of both infrastructure and human lives. The floods can have two type of effects in an area which are namely principal consequence and incidental consequence. The principal consequence of floods incorporate loss of life, destruction of infrastructure, devastation of power generation capacity, scarcity of clean drinking water and increased likelihood of waterborne illness [1]. The incidental consequence comprise of long term effect on the local economy through spread of epidemics, reconstruction and restoration of the infrastructure, dwindling local tourism, lack of food and reestablishment of basic services such as electricity, telecommunication, health, emergency etc.

Floods are mostly caused by increase in river flow due to reasons including heavy rains, melting of snow etc. leading to river overflow and breaking its banks thus flooding the area. Flood and river flow forecasting has been studied by various researchers during the past few decades. Flood forecast models can be classified into the two main categories, physical models and data driven models. Physically models are complicated and need advanced mathematical tools, a significant quantity of calibration data and some degree of expertise and experience with those models [2]. On the other hand, data driven models do not furnish any knowledge of the hydrological processes, they are very helpful for flood forecasting where the primary area of interest is precise predictions of floods [3]. In the recent past, two data driven Soft Computing models that have emerged and became popular in the research community for solving computationally demanding problems are Artificial Neural Networks (ANN) and Support Vector Machine (SVM). These models provide superiority over conventional modeling by providing the ability to handle noisy and uncertain data in dynamic and nonlinear systems thus providing us the ability to utilize them in analyzing and assessing various phenomenon causing disasters where it is not possible to fully avoid the uncertainty in datasets.

In the past few decades, the focus of research has been shifting from conventional methods of forecasting to data driven Soft Computing methods extensively. In [4], a back propagation neural network has been applied for the purpose of flood disaster risk assessment. A neural network based wind prediction method was proposed by authors in [5] for the purpose of power management in northeastern Thailand. In [6], the authors developed a sequential ANN for forecasting river level based on Brahmaputra and Ganges rivers in Bangladesh. The authors have also integrated ANN with Geographic Information System (GIS) in [7] for landslide susceptibility mapping in the Tehri Reservoir region in India. Artificial Neural Networks have also been used by authors of [9] for daily weather forecast in Tiwi, Philippines. In [8], proposed a novel ensemble of ANN and Internet of Things (IoT) employing wireless sensor networks for flood prediction. In [9], the authors compared various configurations of ANN for long range prediction of southwest monsoon rainfall over India. The authors of [10] employed ANN for rainfall runoff modeling and flood prediction.

Recently Support Vector Machine (SVM) has also been gaining importance in this field [11], employs SVM methodology for flood susceptibility mapping in Malaysia. In [12], used SVM in Bayesian context for long term flow forecasting of Kayacik River, Turkey.

The main aim of this study is to analyze the applicability and efficiency of Artificial Neural Networks (ANN) and Support Vector Machine (SVM) for modeling and forecasting floods in the lower Godavari sub-basin located in Eastern India. This paper obtains the results of these two data driven soft computing models and compares them to examine their accuracy in modeling the river flow for flood forecasting and evaluate their performance.

The rest of the paper is organized as follows. Section II describes the various methods implemented in this paper. In Section III, the study area is discussed. Section IV describes the dataset and performance criterion used for testing the accuracy of the various techniques. Section V explains the model inputs, configuration and discusses the results. Section VI concludes the paper.

## II. METHODOLOGY

Data driven Soft Computing methods have been applied by researchers in the field of disaster management successfully as they aim to exploit tolerance is data for imprecision, uncertainty and partial truth to achieve robustness effective solutions [13]. Although it is not possible to fully avoid the natural disasters due to the uncertainty in datasets related to disasters, but their impact can be minimized by developing an appropriate forecasting system, through application of data driven soft computing techniques for more accurate and successful disaster management activities.

### *A. Artificial Neural Networks (ANN)*

Artificial Neural Network is a massively parallel distributed information processing system that has various performance attributes which mimic biological neural networks of the human brain [14]. The aim of an ANN is to generalize a relation of the form of

$$Y^m = f(X^n) \tag{1}$$

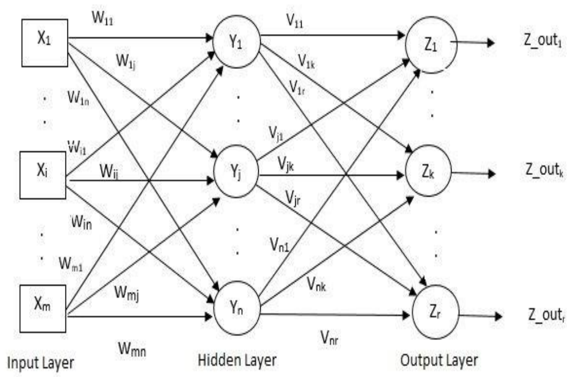



Fig. 1. A multi-layer feed forward neural network with one hidden layer

where $X^n = \{x_1, x_2...,x_n\}$ is an n-dimensional input vector and $Y^m = \{y_1, y_2...,y_m\}$ is an m-dimensional output vector. An artificial neural network is specified by its architecture which exhibit the configuration of connection between nodes, the technique of ascertaining the connection weights and the activation function. The most often used neural network architecture is the multi-layer feed forward structure. A three layered feed forward neural network is composed of numerous units known as neurons or nodes, and connection pathways that link them. The neurons are processing units of the network. A neuron accepts an input signal, processes it, and dispatches an output signal to other neurons connected to it [15]. Fig. 1. shows the architecture of a typical three layered neural network. In hidden and output layers, the total input to $i^th$ neuron is of the form

$$net\_input = \sum_{j=1}^{k} w_{ji}\, y_j + \theta_i \tag{2}$$

where $w_{ji}$ is the weight of neurons i and k is the number of neurons in previous layer. $y_j$ is the output of neuron j and $\theta_i$ is the bias of neuron i. This net input is then passed to an activation or transfer function $f$ to produce the output for neuron i. The Hyperbolic tangent sigmoid transfer function is continuous, monotonically increasing and differentiable function. The hyperbolic tangent sigmoid transfer function for neuron i is of the form

$$output = \frac{2}{(1+e^{-2n})-1} \tag{3}$$

ANN also requires a training algorithm to learn and solve a problem. Levenberg Marquardt backpropagation and Bayesian Regularization backpropagation learning algorithms are used very often as they provide better performance with good learning speeds.

### *B. Support Vector Machine (SVM)*

SVM is a neural network technology based on statistical learning. They were designed based on statistical learning theory and were obtained from the structural risk minimization hypothesis to diminish both empirical risk and the confidence interval of the learning machine so as to achieve a good generalization capability [16]. The configuration of an SVM is not decided before. The input vectors sustaining the model configuration are determined through a model training procedure is described below.

Consider a linear regression problem trained on the dataset $\chi = \{u_i,v_i;i = 1,2,...,n\}$ with input vectors $u_i$ and target vectors $v_i$, then the regression function of SVM can be given as:

$$f(u) = w_i \cdot \varphi_i(u) + b \tag{4}$$

where $w_i$ is the weight vector, b is the bias and $\varphi_i$ is a nonlinear transfer function that maps the input to a high dimensional feature space in which a linear regression can manage with nonlinear regression of the input space. Vapnik in [17] introduced the following convex optimization problem with -insensitive loss function for obtaining the solution to Eq. 5:

$$Minimize:\ \frac{1}{2}\,.\,\|w^2\| + C\left(\sum_{i}^{N}(\xi_i + \xi_i^*)\right) \tag{5}$$

$$Subject\ to \begin{cases} w_i.\phi(u_i) + b_i - v_i \le \epsilon + \xi_i^*, for\ i = 1,2,\dots,N \\ v_i - w_i.\phi(u_i) - b_i \le \epsilon + \xi_i, for\ i = 1,2,\dots,N \\ \xi_i, \xi_i^*, \qquad for\ i = 1,2,\dots,N \end{cases} \tag{6}$$

where $\xi_i$ and $\xi_i*$ are slack variables introduced to assess the divergence of training vectors outside -insensitive zone and C is a positive constant regulating the degree of penalizing loss when a training error occurs. Eq. 6 is solved by using Lagrangian multipliers and applying the Karush-Kuhn-Tucker (KKT) optimality condition. Thus, the input vectors which have nonzero Lagrangian multipliers under the KKT condition support the structure of the estimator and are called support vectors.

## III. STUDY AREA

The applicability of the various mentioned techniques as a time series forecasting model is studied in this paper. To demonstrate the ability and validity of these methods for time series forecasting and modeling, the Godavari river, the biggest in eastern India is chosen. The river has been used for irrigation,

TABLE I STATISTICAL PARAMETERS OF THE DATASETS.

| | $M_{min}$ | $M_{max}$ | $M_{mean}$ | $M_{stdev}$ | $M_{ske}$ |
|---|---|---|---|---|---|
| Training | 34 | 47248.87 | 2931.43 | 5674.60 | 3.36 |
| Test | 154.8 | 827.01 | 340.14 | 115.98 | 0.63 |

domestic and industrial use and hydropower generation. The Cauvery river basin is one of the most important agricultural regions in east India. It has a length of 1465 km and a drainage area of 312,813 $km^2$. The annual runoff potential of Godavari river is 110.54 $km^3$ [18]. The location of Godavari River and its drainage basin are shown on Fig. 2. There are two river flowing gauge stations, Bhadrachalam and Polavaram equipped with automatic daily flow recorders on the Godavari river main branch as shown in Fig 2. As can be seen in the figure, the river flow gauging station of Bhadrachalam is located upstream of Polavaram. The data records of both these river gauging stations are used for river flow and flood forecast modeling.

## IV. MODEL DEVELOPMENT

### A. Description of Data

In this paper, the performance of ANN and SVM were examined on daily flow. To achieve this, 8 year flow data was available from 2007 to 2015 [19]. In total, the number of days for which the flow data was available were 2630. The data were divided into two sets: a training data composed of years 20072014 and a testing dataset of year 2015. This paper utilizes whole year dataset was used in training of the models as it allows the

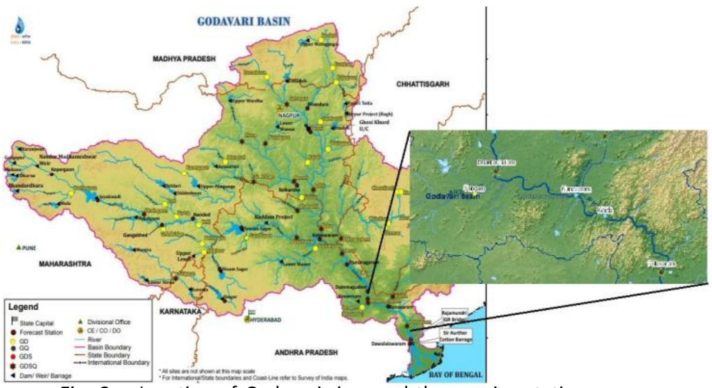



Fig. 2. Location of Godavari river and the gauging stations

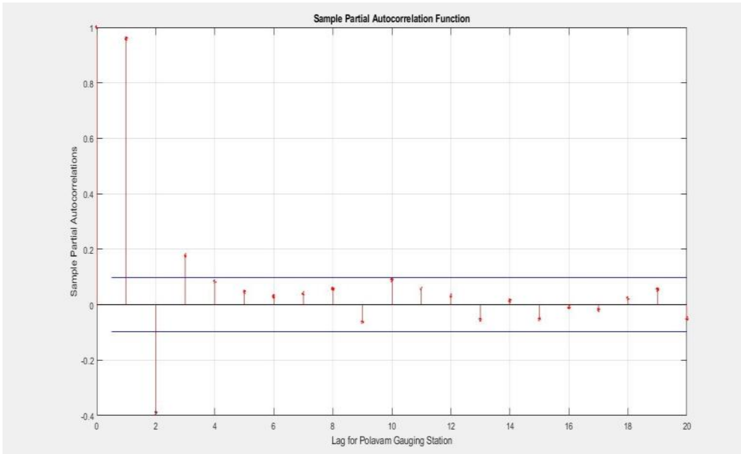
incorporation of numerous hydrological conditions that are prevalent throughout different seasons of the year. In this way, the models become more resilient for handling different hydrological conditions that occur in the whole time series [20]. The daily statistical parameters which contain the minimum value $M_{min}$, maximum value $M_{max}$, mean $M_{mean}$, standard deviation $M_{stdev}$ and skewness coefficient $M_{ske}$ of the river flow data are shown on Table I.

The number of lags were selected according to the partial auto-correlation function (PCF) of daily flow data of Polavaram gauging station which is shown in Fig. 3.It is clear from the figure that first three lags have significant effects on $M_{t+1}$. The cross correlation of the Polavaram and Bhadrachalam gauging stations presented in Fig. 4. shows a significant correlation for up to three days lag in the flow data. Thus three previous lags of Polavaram and first, third and fourth lags of Bhadrachalam gauging stations were considered as inputs to the model in this

TABLE II
MODEL STRUCTURES FOR FORECASTING.

| Model No. | Input Structure | No. of Variables | Output |
|---|---|---|---|
| M1 | $P_{t-1}$ | 1 | $P_{t+1}$ |
| M2 | $P_{t-1}P_{t-2}$ | 2 | $P_{t+1}$ |
| M3 | $P_{t-1}P_{t-2}P_{t-3}$ | 3 | $P_{t+1}$ |
| M4 | $P_{t-1}B_{t-1}$ | 2 | $P_{t+1}$ |
| M5 | $P_{t-1}P_{t-2}B_{t-1}$ | 3 | $P_{t+1}$ |
| M6 | $P_{t-1}P_{t-2}P_{t-3}B_{t-1}$ | 4 | $P_{t+1}$ |
| M7 | $P_{t-1}B_{t-3}$ | 2 | $P_{t+1}$ |
| M8 | $P_{t-1}P_{t-2}B_{t-3}$ | 3 | $P_{t+1}$ |
| M9 | $P_{t-1}P_{t-2}P_{t-3}B_{t-3}$ | 4 | $P_{t+1}$ |
| M10 | $P_{t-1}B_{t-4}$ | 2 | $P_{t+1}$ |
| M11 | $P_{t-1}P_{t-2}B_{t-4}$ | 3 | $P_{t+1}$ |
| M12 | $P_{t-1}P_{t-2}P_{t-3}B_{t-4}$ | 4 | $P_{t+1}$ |
| M13 | $P_{t-1}B_{t-1}B_{t-3}$ | 3 | $P_{t+1}$ |
| M14 | $P_{t-1}B_{t-1}B_{t-3}B_{t-4}$ | 4 | $P_{t+1}$ |
| M15 | $P_{t-1}P_{t-2}B_{t-1}B_{t-3}$ | 4 | $P_{t+1}$ |
| M16 | $P_{t-1}P_{t-2}B_{t-1}B_{t-3}B_{t-4}$ | 5 | $P_{t+1}$ |
| M17 | $P_{t-1}P_{t-2}P_{t-3}B_{t-1}B_{t-3}$ | 5 | $P_{t+1}$ |
| M18 | $P_{t-1}P_{t-2}P_{t-3}B_{t-1}B_{t-3}B_{t-4}$ | 6 | $P_{t+1}$ |

Fig. 3. Partial auto-correlation function of daily flow data of Polavaram Gauge station.




study. The inputs present the previous flow (t-1,t-2,t-3 and t-4) and the output corresponds to the flow at time t+1. Thus, the structure of the forecasting models are shown in Table II where the Polavaram gauge flow data is represented as *P* and Bhadrachalam gauge flow data is represented as *B*

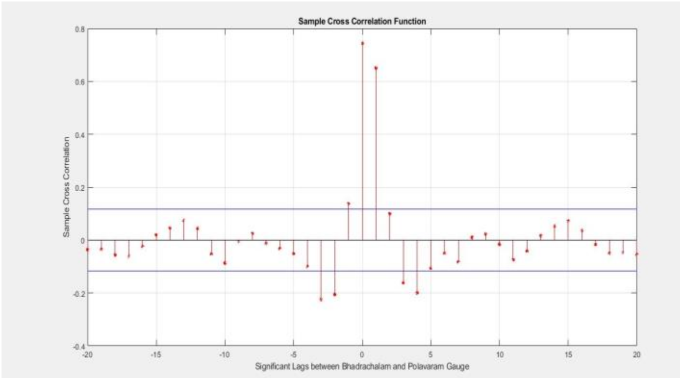



Fig. 4. Cross-correlation function of daily flow data of Polavaram and Bhadrachalam Gauge stations.

*B. Data Pre-processing*

For the purpose of obtaining efficient and accurate training of the models, the data are needed to be normalized. It was reported in [21] that models trained on normalized data attain better performance and rapid convergence. In this paper, normalization is performed on all data scaled in the range 0 1 independently by employing the following equation:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{7}$$

where $X$ is the normalized value, $X$ is the sample value, $X_{min}$ is the minimum value and $X_{max}$ is the maximum value.

*C. Model Performance Criteria*

The performance of the models developed in this paper were evaluated using three standard statistical performance assessment criteria. The statistical measures used were the Mean Squared Error (MSE), coefficient of correlation also known as Regression (R) and Nash-Sutcliffe efficiency coefficient (NS). MSE gives the information about the predictive ability of the model, R measures the degree to which two variables are linearly related and NS gives the predictive power of the models. MSE provides the average squared difference between output of the model and the actual test outputs. It can be calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(t_i - a_i)^2 \tag{8}$$

where $n$ is the size of dataset, $a_i$ is the output of the model and $t_i$ is the corresponding actual output.

R is defined as the correlation between targets and outputs. When the value of R = 1, it means that there is a close relationship between targets and outputs and is R = 0, it means that there is random relationship between the two. It is calculated by the equation:

$$R = \frac{\sum_{i=1}^{n}(t_i - \bar{t})(a_i - \bar{a})}{\sqrt{\sum_{i=1}^{n}(t_i - \bar{t})^2 \sum_{i=1}^{n}(a_i - \bar{a})^2}} \tag{9}$$

The Nash-Sutcliffe efficiency coefficient (NS) can be measured as:

$$NS = 1 - \frac{\sum_{i=1}^{n}(t_i - a_i)^2}{\sum_{i=1}^{n}(t_i - \bar{t})^2} \tag{10}$$

where $n$ is the size of dataset, $a_i$ is the output of the model and $t_i$ is the corresponding actual output.A model can be claimed to give a perfect prediction if the NS criterion is equal to 1 but a model can considered as accurate if the NS value of larger than 0.8 as shown in [22].

## V. RESULTS AND DISCUSSION

In this study ANN and SVM methods were applied to the models developed above and the results are described in this section. The implementation and analysis of results of the above mentioned techniques were performed in MATLAB 2017b.

*A. Artificial Neural Networks (ANN)*

In this study, two back propagation training algorithms - the Levenberg-Marquardt (LM) and Bayesian Regularization (BR) were used to to train the ANN models. The performance of these two algorithms was then compared mainly in terms of Mean Squared Error (MSE), Correlation Coefficient (R) and Nash-Sutcliffe coefficient (NS). Each model was trained using different number of hidden neurons (HN) in the hidden layer. Tables III and IV present the testing results obtained from the eighteen models trained by LM and BR algorithms respectively. As can be observed from tables III and IV that Model 1 with 10 hidden neurons and trained with the Bayesian Regularization algorithm composed of one antecedent flow data of Polavaram gauging station is chosen as best fit model for ANN as it has lowest MSE value of 0.00509 , highest R value of 0.911 and highest value of NS 0.8282 during testing. The comparison between the observed and the ANN computed temporal variation of flow obtained during testing of Model 1 is shown in Fig. 5.

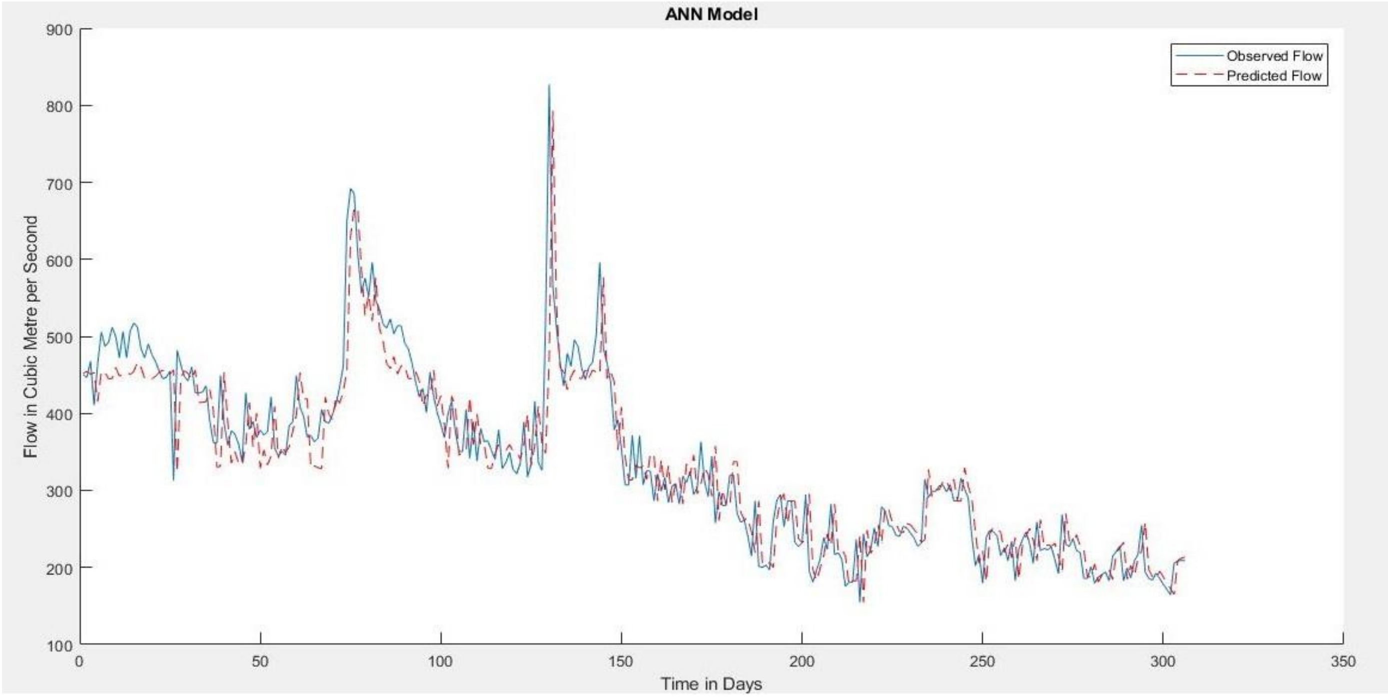



Fig. 5. Comparison of observed and ANN predicted flow obtained from Model 1 with 10 hidden neurons and trained with Bayesian Regularization algorithm during testing.

*B. Support Vector Machine (SVM)*

The SVM technique was applied to all eighteen models and the performance indices obtained were compared in Table V. As can be seen from the table that the Model 1 which consists of both antecedent flow data of Musiri gauging station and the second antecedent flow data of Kodumudi gauging station has the lowest MSE value of 0.00473, highest R value of 0.923 and the highest NS value of 0.8405 thus showing the best fit for SVM method. The comparison between the observed and the SVM computed temporal variation of flow obtained during testing of Model 1 is shown in Fig. 6.

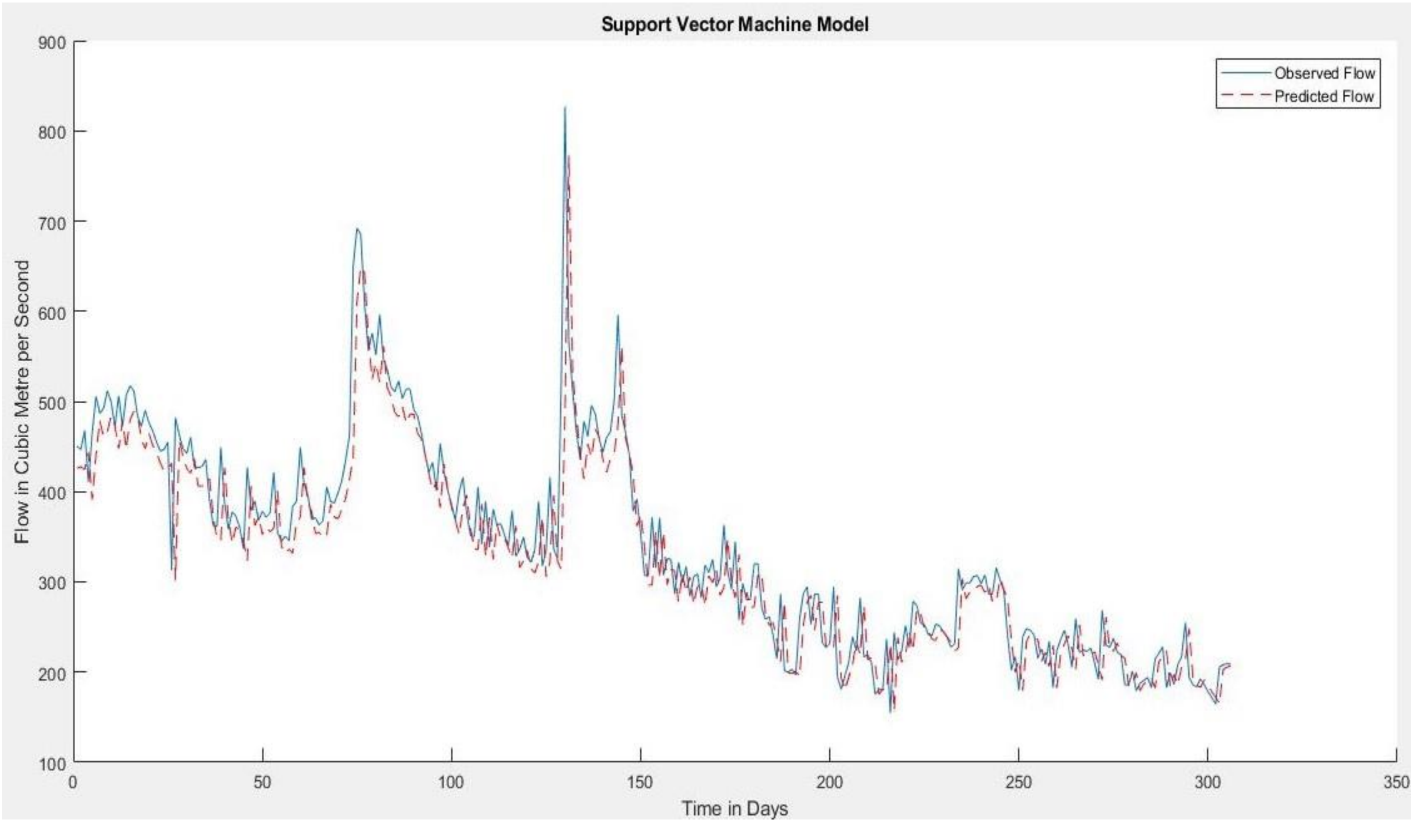



Fig. 6. Comparison of observed and SVM predicted flow obtained from Model 1 during testing.

TABLE III

PERFORMANCE INDICES OF ANN MODELS USING THE LEVENBERG-MARQUARDT ALGORITHM

| Model | HN | MSE | R | NS | HN | MSE | R | NS | HN | MSE | R | NS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M1 | 10 | 0.00526 | 0.908 | 0.8226 | 20 | 0.00532 | 0.906 | 0.8206 | 30 | 0.00672 | 0.882 | 0.7734 |
| M2 | 10 | 0.00594 | 0.895 | 0.7997 | 20 | 0.00750 | 0.869 | 0.7474 | 30 | 0.00724 | 0.873 | 0.7562 |
| M3 | 10 | 0.25782 | 0.645 | 0.0791 | 20 | 0.58487 | 0.398 | 0.0881 | 30 | 0.15348 | 0.088 | 0.1666 |
| M4 | 10 | 0.02669 | 0.688 | 0.1006 | 20 | 0.05495 | 0.558 | 0.1512 | 30 | 0.05826 | 0.595 | 0.4628 |
| M5 | 10 | 0.35438 | 0.225 | 0.0348 | 20 | 0.19021 | 0.634 | 0.1061 | 30 | 0.84527 | 0.167 | 0.0067 |
| M6 | 10 | 0.02649 | 0.709 | 0.1080 | 20 | 0.02098 | 0.718 | 0.2934 | 30 | 0.03665 | 0.587 | 0.2338 |
| M7 | 10 | 0.43566 | 0.692 | 0.0655 | 20 | 0.09723 | 0.744 | 0.1209 | 30 | 0.84554 | 0.142 | 0.0046 |
| M8 | 10 | 0.57928 | 0.361 | 0.0051 | 20 | 0.39341 | 0.561 | 0.0631 | 30 | 0.23186 | 0.596 | 0.1065 |
| M9 | 10 | 1.77999 | 0.014 | 0.0018 | 20 | 0.44747 | 0.470 | 0.0434 | 30 | 2.53031 | 0.018 | 0.0016 |
| M10 | 10 | 0.10642 | 0.500 | 0.5780 | 20 | 0.80701 | 0.155 | 0.0084 | 30 | 0.26675 | 0.325 | 0.1008 |
| M11 | 10 | 0.05629 | 0.594 | 0.1020 | 20 | 0.10651 | 0.387 | 0.1809 | 30 | 0.10645 | 0.327 | 0.1562 |
| M12 | 10 | 0.01993 | 0.788 | 0.3296 | 20 | 0.07792 | 0.647 | 0.2731 | 30 | 0.29393 | 0.582 | 0.0923 |
| M13 | 10 | 1.43762 | 0.242 | 0.0039 | 20 | 0.46944 | 0.251 | 0.0425 | 30 | 1.44918 | 0.157 | 0.0089 |
| M14 | 10 | 0.03648 | 0.219 | 0.1005 | 20 | 0.06294 | 0.528 | 0.0325 | 30 | 0.16749 | 0.519 | 0.1313 |
| M15 | 10 | 0.02584 | 0.584 | 0.1301 | 20 | 0.02621 | 0.663 | 0.1174 | 30 | 0.03486 | 0.474 | 0.1738 |
| M16 | 10 | 0.02528 | 0.682 | 0.1498 | 20 | 0.02491 | 0.633 | 0.1625 | 30 | 0.02078 | 0.723 | 0.3012 |
| M17 | 10 | 0.02709 | 0.717 | 0.0879 | 20 | 0.02118 | 0.667 | 0.2868 | 30 | 0.03771 | 0.509 | 0.2697 |
| M18 | 10 | 0.19355 | 0.645 | 0.1074 | 20 | 0.19355 | 0.534 | 0.1074 | 30 | 0.22058 | 0.588 | 0.1162 |

TABLE IV

PERFORMANCE INDICES OF ANN MODELS USING THE BAYESIAN REGULARIZATION ALGORITHM

| Model | HN | MSE | R | NS | HN | MSE | R | NS | HN | MSE | R | NS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M1 | 10 | 0.00509 | 0.911 | 0.8282 | 20 | 0.00661 | 0.883 | 0.7772 | 30 | 0.01773 | 0.748 | 0.4025 |
| M2 | 10 | 0.00510 | 0.910 | 0.8279 | 20 | 0.00543 | 0.905 | 0.8168 | 30 | 0.10354 | 0.229 | 0.3872 |
| M3 | 10 | 0.00546 | 0.904 | 0.8159 | 20 | 0.09553 | 0.606 | 0.2158 | 30 | 0.48237 | 0.137 | 0.2378 |
| M4 | 10 | 0.16821 | 0.143 | 0.6667 | 20 | 1.47042 | 0.282 | 0.1341 | 30 | 0.85502 | 0.566 | 0.1256 |
| M5 | 10 | 3.83358 | 0.129 | 0.1062 | 20 | 0.07702 | 0.652 | 0.5939 | 30 | 0.21581 | 0.221 | 0.2683 |
| M6 | 10 | 0.06993 | 0.435 | 0.3543 | 20 | 0.40169 | 0.304 | 0.5221 | 30 | 0.40499 | 0.443 | 0.6330 |
| M7 | 10 | 0.21649 | 0.704 | 0.2876 | 20 | 0.20458 | 0.520 | 0.5886 | 30 | 0.24208 | 0.653 | 0.4190 |
| M8 | 10 | 1.79870 | 0.435 | 0.1486 | 20 | 0.29623 | 0.404 | 0.5184 | 30 | 0.97548 | 0.062 | 0.1372 |
| M9 | 10 | 1.56484 | 0.206 | 0.2764 | 20 | 4.53991 | 0.256 | 0.0040 | 30 | 9.80816 | 0.018 | 0.0036 |
| M10 | 10 | 0.48207 | 0.692 | 0.2075 | 20 | 0.16007 | 0.775 | 0.3818 | 30 | 2.87071 | 0.184 | 0.1550 |
| M11 | 10 | 0.66192 | 0.490 | 0.2542 | 20 | 0.69184 | 0.641 | 0.2603 | 30 | 6.50270 | 0.154 | 0.0241 |
| M12 | 10 | 0.86897 | 0.788 | 0.2153 | 20 | 2.52424 | 0.334 | 0.0664 | 30 | 5.13709 | 0.253 | 0.1115 |
| M13 | 10 | 0.57192 | 0.447 | 0.2522 | 20 | 0.06174 | 0.542 | 0.5843 | 30 | 0.39573 | 0.886 | 0.3213 |
| M14 | 10 | 0.19085 | 0.245 | 0.4166 | 20 | 0.90205 | 0.255 | 0.3274 | 30 | 0.21987 | 0.583 | 0.3922 |
| M15 | 10 | 0.13428 | 0.075 | 0.5202 | 20 | 0.89484 | 0.220 | 0.3290 | 30 | 1.01358 | 0.204 | 0.1198 |
| M16 | 10 | 0.13999 | 0.225 | 0.2066 | 20 | 0.86610 | 0.320 | 0.3488 | 30 | 0.91992 | 0.426 | 0.1283 |
| M17 | 10 | 0.69911 | 0.322 | 0.3621 | 20 | 0.47806 | 0.564 | 0.5467 | 30 | 0.63242 | 0.537 | 0.2888 |
| M18 | 10 | 0.54615 | 0.383 | 0.3651 | 20 | 3.91364 | 0.199 | 0.0087 | 30 | 1.69002 | 0.213 | 0.1136 |

The performances of best fit models of ANN and SVM are shown in Table VI. It can be observed from the results that SVM model seem to perform better than other models as it has minimum MSE and highest R and NS values. Both models showed good prediction for low values of flow but were able to maintain their accuracy for peak flow and both underestimated the peak flow.Overall, the SVM and ANN techniques can give good forecasting performance and could be successfully employed to establish prediction models that could provide accurate and reliable flood forecasts. The results show that the SVM model was superior to other models in flood and river flow forecasting.

TABLE V

PERFORMANCE INDICES OF SVM MODELS

| Model | MSE | R | NS |
|---|---|---|---|
| M1 | 0.00473 | 0.923 | 0.8405 |
| M2 | 0.00569 | 0.905 | 0.8083 |
| M3 | 0.15311 | 0.805 | 0.1542 |
| M4 | 0.01463 | 0.786 | 0.5069 |
| M5 | 0.06104 | 0.800 | 0.0558 |
| M6 | 0.01153 | 0.819 | 0.6117 |
| M7 | 0.20878 | 0.873 | 0.0282 |
| M8 | 0.07367 | 0.875 | 0.4802 |
| M9 | 0.07241 | 0.900 | 0.4375 |
| M10 | 0.04293 | 0.798 | 0.4434 |
| M11 | 0.00757 | 0.877 | 0.7453 |
| M12 | 0.00755 | 0.874 | 0.7461 |
| M13 | 0.14845 | 0.841 | 0.3373 |
| M14 | 0.03271 | 0.576 | 0.0998 |
| M15 | 0.00974 | 0.840 | 0.6719 |
| M16 | 0.00965 | 0.842 | 0.6754 |
| M17 | 0.00870 | 0.854 | 0.7070 |
| M18 | 0.01796 | 0.707 | 0.3961 |

TABLE VI: COMPARISON OF THE PERFORMANCES OF ANN AND SVM BEST FIT MODELS

| Technique | Model | MSE | R | NS |
|---|---|---|---|---|
| ANN | M1 | 0.00509 | 0.911 | 0.8282 |
| SVM | M1 | 0.00473 | 0.923 | 0.8405 |

## VI. CONCLUSION

In this study, ANN and SVM models were developed for forecasting of floods based on antecedent values of river flow data. For attaining the objective, the Polavaram and Bhadrachalam gauging stations located on the Godavari River in eastern India has been selected as case study. The results of ANN and SVM models were compared and evaluated based on their testing performance. While comparing the results of these models it was observed that the MSE values of SVM model was lowest among both models. Moreover, the R and NS values of SVM model were higher than those of ANN model. Therefore, the SVM model could improve the accuracy over the ANN model. The results also demonstrated that both models showed good forecast accuracy for low values of flow but were unable to maintain their accuracy for peak flow value as both ANN and SVM models underestimated the peak flow. Overall, the analysis done in this study demonstrates that SVM method was better to the other methods in flood forecasting.

## VII. REFERENCES


[1]. P. Nayak, K. Sudheer, D. Rangan, and K. Ramasastri, “Short-term flood forecasting with a neurofuzzy model,” Water Resources Research, vol. 41, no. 4, 2005.

[2]. Q. Jin, J. Chen, H. Wang, and S. Zhao, “Research on regional flood disaster risk assessment based on pca and bp neural network,” in EProduct E-Service and E-Entertainment (ICEEE), 2010 International Conference on. IEEE, 2010, pp. 1–4.

[3]. C. Surussavadee and W. Wu, “A neural network-based wind forecasting model for wind power management in northeastern thailand,” in Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. IEEE, 2015, pp. 3957–3960.

[4]. M. S. A. Siddiquee and M. M. A. Hossain, “Development of a sequential artificial neural network for predicting river water levels based on brahmaputra and ganges water levels,” Neural Computing and Applications, vol. 26, no. 8, pp. 1979–1990, 2015.

[5]. R. Kumar and R. Anbalagan, “Remote sensing and gis based artificial neural network system for landslide suceptibility mapping,” in Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. IEEE, 2015, pp. 4696–4699.

[6]. K. L. M. D. Sobrevilla, A. G. Quinones, K. V. S. Lopez, and V. T.˜Azana, “Daily weather forecast in tiwi, albay, philippines using artificial˜ neural network with missing values imputation,” in Region 10 Conference (TENCON), 2016 IEEE. IEEE, 2016, pp. 2981–2985.

[7]. P. Mitra, R. Ray, R. Chatterjee, R. Basu, P. Saha, S. Raha, R. Barman, S. Patra, S. S. Biswas, and S. Saha, “Flood forecasting using internet of things and artificial neural networks,” in Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual. IEEE, 2016, pp. 1–5.

[8]. S. Chakraverty and P. Gupta, “Comparison of neural network configurations in the long-range forecast of southwest monsoon rainfall over india,” Neural Computing and Applications, vol. 17, no. 2, pp. 187–192, 2008.

[9]. M. S. Tehrany, B. Pradhan, and M. N. Jebur, “Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in gis,” Journal of hydrology, vol. 512, pp. 332–343, 2014.

[10]. U. Okkan, Z. A. Serbes, and P. Samui, “Relevance vector machines approach for long-term flow prediction,” Neural Computing and Applications, vol. 25, no. 6, pp. 1393–1405, 2014.

[11]. S. K. Pal and A. Ghosh, “Soft computing data mining,” 2004.

[12]. S. Haykin, “Neural networks a comprehensive foundation: Pretice hall international,” Inc., Englewood Cliffs, 1999.

[13]. Z. He, X. Wen, H. Liu, and J. Du, “A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region,” Journal of Hydrology, vol. 509, pp. 379–386, 2014.

[14]. P. C. Deka et al., “Support vector machine applications in the field of hydrology: a review,” Applied Soft Computing, vol. 19, pp. 372–386, 2014.

[15]. V. Vapnik, “The nature of statistical learning theory springer new york google scholar,” 1995.

[16]. S. Bhawan and R. Puram, “Watershed atlas of india,” 2014.

[17]. I. WRIS, “India-wris webgis water resource information system of india, http://www.india-wris.nrsc.gov.in,” 2017. [Online]. Available: http://www.india-wris.nrsc.gov.in

[18]. O. Kis¸i, “Daily pan evaporation modelling using a neuro-fuzzy computing¨ technique,” Journal of hydrology, vol. 329, no. 3-4, pp. 636–646, 2006.

[19]. M. Shanker, M. Y. Hu, and M. S. Hung, “Effect of data standardization on neural network training,” Omega, vol. 24, no. 4, pp. 385–397, 1996.

[20]. C. Shu and T. Ouarda, “Regional flood frequency analysis at ungauged sites using the adaptive neuro-fuzzy inference system,” Journal of Hydrology, vol. 349, no. 1-2, pp. 31–43, 2008.


### Authors Profile

*P. Misra* is an Assistant Professor (Stage-III) of Computer Science in the Department of Computer Science at the University of Lucknow, Lucknow, U.P., India. He received bachelor degree (1995) in Physics and Maths and a dual Master’s degree in Electronics and Computer Applications, and a Ph.D. degree (2003) from the University of Lucknow. He has published more than 45 research papers in reputed international and conferences including IEEE and it’s also available online. He is currently engaged in research areas which includes Machine Learning, Soft computing, Artificial Intelligent Systems, human-computer interaction, Sensor fabrication & applications and issues related to cybercrime and its prevention policies etc.

*Mr S. Shukla* pursed Bachelor of Technology for Jaypee University of Information Technology, India in the year 2010 and Master of Technology from Manipal Institute of Technology, India in year 2013. He is currently pursuing Ph.D. from Department of Computer Science, University of Lucknow. He has published more than 8 research papers in reputed international and conferences including IEEE and it’s also available online. His main research work focuses on Soft Computing, Machine Learning and Artificial Intelligence. He has 4 years of Research Experience.