CHAPTER 5

RIVER WATER LEVEL IDENTIFICATION USING RADIAL BASIS FEED FORWARD NETWORK

5.1 INTRODUCTION

Recently, the impacts of nature caused by rivers generate huge damage to the society. Certain precautions can be taken when the disasters are forecast and hence forecasting of river stages may provide a solution to this problem. In order to accomplish this objective, a mechanism to determine the variations of the river water level and predict the stage of the water through satellite images using the radial basis feed forward network is discussed in this chapter. This work comprises three phases namely preprocessing phase, training phase and testing phase. The satellite river image is preprocessed and analyzed to identify the river water regions using the designed Radial Basis Feed Forward Neural Network (RBFNN). It is then used to predict the water level in the river using another RBFNN. In the testing phase, the input satellite river image is tested and the stage of the river, whether it is in flooding, normal or drought stage is identified.

5.2 RADIAL BASIS FEED FORWARD NETWORK

Radial basis function networks (RBFNs) compute activations at the hidden neurons in a way that is different from the case of feed forward networks. Rather than employing an inner product between the input vector and the weight vector, hidden neuron activations in RBFNs are computed
using an exponential variant of a distance measure (usually the Euclidean distance or a weighted norm) between the input vector and a prototype vector that characterize the signal function at a hidden neuron.

A radial basis function (RBF) is a real-valued function whose value depends only on the distance from the origin. If a function ‘h’ satisfies the property \( h(x) = h(\|x\|) \), then it is a radial function. Their characteristic feature is that, their response decreases (or increases) monotonically with a distance from a central point. The centre, the distance scale, and the precise shape of the radial function are a few parameters of the model, all fixed if it is linear.

A typical radial function is the Gaussian Equation (5.1) which, in the case of a scalar input, is,

\[
h(x) = \exp\left(\frac{-(x-c)^2}{r^2}\right) \tag{5.1}
\]

Its parameters are its center \( c \) and its radius \( r \).

A Gaussian RBF monotonically decreases with distance from the center. In contrast, a multi-quadric RBF which, in the case of scalar input, monotonically increases with distance from the center. Gaussian-like RBFs are local (given a significant response only in a neighborhood near the center) and are more commonly used than multi-quadric type RBFs which have a global response. Radial functions are simply a class of functions. In principle, they could be employed in any sort of model (linear or nonlinear) and any sort of network (single-layer or multi-layer). RBF networks have traditionally been associated with radial functions in a single-layer network. The input layer carries the outputs of basis function. The distance between these values and center values are found and summed up form linear combination before the neurons of the hidden layer. These neurons are said to contain the radial basis function with exponential form and the outputs of the RBF activation
function are further processed according to specific requirements (Sambasiva Rao 2008). An example of the RBF network is shown in Figure 5.1.

![Radial Basis Function Network](image)

**Figure 5.1 Radial Basis Function Network**

5.3 PREPROCESSING PHASE

In this phase, the input satellite image is analyzed and segmented to train the RBFNN. Initially, the input image is filtered to sluice the noises and then the denoised river image is put into further process. The denoising process has been carried out to acquire the acute information about the river. The watered region of the river must be identified because the image is a satellite image and it not only wraps the watered area it wraps up the land area also. Hence it is important to identify the watered regions in the image. In order to identify the watered region, the image must be preprocessed. Let $D$ be the database of river images and let the dimension of each image $I$ be $M \times N$ and $p_{ij}$ be the number of pixels of the image. Initially, the water regions of the database river images are selected manually from the collection of images that exist in the database $D$ and then the pixels of these water regions are given to the RBFNN for the training process. Before commencing the identification of watered regions, the preprocessing steps have been taken.
\[
D_k = \{I_0, I_1, I_2, \ldots, I_{N_j-1}\} \quad 0 \leq k \leq N_j - 1 \quad (5.2)
\]

Let \( I_{\text{imp}} \) be one of the images from the database \( D \) to be analyzed. It is of dimension \( M \times N \) which is of \( p_i \) pixels. Initially, the image is filtered to remove the noises from it and then the denoised image is converted to the LAB color space which is utilized to identify the acute information from the image. A Lab color space is a color-opponent space with dimension \( L \) for lightness and \( a \) and \( b \) for the color-opponent dimensions, based on nonlinearly compressed CIE XYZ color space coordinates. The following Equations (5.3) to (5.6) detail the process:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
0.412453 & 0.357580 & 0.180423 \\
0.212671 & 0.715160 & 0.072167 \\
0.019334 & 0.119193 & 0.950227
\end{bmatrix} \begin{bmatrix}
r \\
g \\
b
\end{bmatrix}
\]  

\[
L = \begin{cases}
116 * (Y / Y_n)^{1/3} - 16, & \text{for } Y / Y_n > 0.008856 \\
903.3 * Y / Y_n, & \text{otherwise}
\end{cases}
\]  

\[
a = 500 * ( f (X / X_n) - f (Y - Y_n))
\]

\[
b = 200 * ( f (Y / Y_n) - f (Z - Z_n))
\]

where \( f(t) = \begin{cases}
t^{1/3}, & \text{for } t > 0.008856 \\
7.787 * t + 16 / 116, & \text{otherwise}
\end{cases}
\]  

(Hoffman 2000)

Here \( X_n, Y_n, \) and \( Z_n \) are the tri-stimulus values of the reference white. Hence the denoised image which is in RGB color space is converted to LAB color space. Subsequent to the aforesaid process, the LAB color spaced image is segmented by means of non-linear diffusion. The input image is shown in Figure 5.2.
5.3.1 Segmentation

Initially the input river image \( I_{\text{inp}} \) is subjected to the segmentation process and the segmented image shown in Figure 5.3. Clustering pixels of the image into prominent image regions i.e. areas that signify individual surfaces, objects, or natural parts of the objects, is the objective of image segmentation. To accomplish this segmentation process, the non-linear diffusion is applied on the \( I_{\text{inp}} \) image to get the segmented image \( I_{\text{sgm}} \).
5.3.2 Binarization

The segmented image $I_{sgm}$ is then converted to the binary image which is then utilized for further processes. In this binarization process, the converted LAB color spaced image is given as the input and the following pseudo-code details the conversion process:

$$I_{gr} = \frac{(L+a+b)_{sgm}}{3}$$

For $\forall p_{ij} \in I_{gr}$

If $p_{ij} \geq thr$

Set $p_{ij}=1$

Else

$p_{ij}=0$

End if

End for

End for

In this pseudo-code $thr$ is the global threshold parameter. Hence the converted binary image $I_{bary}$ is obtained. Figure 5.4 reveals the binary image.
5.3.3 Applying Morphological Operation

Here, the Morphological operation which plays a significant role in machine vision and automatic object detection to recognize the structure or form of an image is applied. This generally detects objects or boundaries present in an image. The binary image \( I_{bny} \) is applied with the morphological operation. Here the ‘imclose’ morphological operation which performs morphological closing on the grayscale or binary image returning the closed image is utilized. Figure 5.5 displays the morphological image.
5.3.4 Thresholding

Satellite images not only cover the watered area they also cover the land area and so the images contain the small watered regions which are not the river water regions. In order to avoid the misunderstanding of this small watered area, thresholding operation has been done on the image and then the altered image is obtained. The resulting image is depicted in Figure 5.6.
5.3.5 Suppression

In this step, the surplus regions in the image are suppressed and the remaining water regions are taken for consideration. The resulting suppressed image $I_{spr}$ is utilized to analyze the water level changes of the river. After this process, the regions are identified. $I_{spr}$ is the suppressed image and then these regions are marked in the original image $I_{inp}$. Next, these image regions are given as the input to train the RBFN network which is utilized to identify the river watered region. The detailed process is explained in the subsequent section. The water region marked image is shown in Figure 5.7.

![Figure 5.7 Water Regions Marked in the Original Image](image)

5.4 TRAINING PHASE

5.4.1 Training Phase for Water Region Identification

The preprocessed image is used to train the RBFNN. The proposed technique is integrated with the training process to identify the river watered regions. In order to accomplish this the RBFN network is utilized. The network is characterized by a set of inputs and a set of outputs and the
Gaussian function is embedded into the two-layer feed forward neural network. In between the inputs and outputs there is a layer of processing units, called hidden units. Each of them implements the Gaussian function. The way in which the network is used for data modeling is different when approximating time-series and in pattern classification. The river water regions are identified in this training process and for that the identified river water regions are given as the input.

In this network the water regions are processed pixel by pixel and the R, G, and B values of a pixel in the selected water regions are given as input to the neural network. This network maps 3 input values to one output value and this can be thought of as the graphical representation of a parametric function. The water regions presented in the river images which are selected manually are utilized on this network to train this RBFN network with Bayesian regulation back propagation. For the purpose of this network, a training dataset is generated via pixels which are collected from the water region of river images. RBFN network is generated, and trained with the collected pixel of the water region dataset. It outputs that the selected pixel is watered region or not. Thus, the network is configured with 3 input units; \( N_h \) hidden units and one output unit as shown in Figure 5.8. The number of hidden layers and the number of neurons are varied and the results are manifested in Tables 5.1 and 5.2.
The following steps are adapted to design RBFN:

**Step 1:** Initially, the input weights are set to every neuron, apart from the neurons of the input layer.

**Step 2:** RBFN network is designed with 3 input layers which are \( R, G \) and \( B \) pixel values of the regions which are selected as watered regions. \( N_h \) hidden layers and one output layer which indicates that the pixel from the appropriate region is watered or not. The value at the output layer is either true or false depending on whether the input pixel belongs to a water region or not.

**Step 3:** A neural network is designed with 3 input layers which are the \( R, G \) and \( B \) pixel values of the selected water regions, \( N_h \) hidden layers and one output layer to show whether the selected pixel belongs to the water region or not. The value at the output layer is either true or false depending on whether the input pixel belongs to a water region or not. In this neural network, 3 input neurons and a bias neuron, \( N_h \) hidden neurons and a bias neuron and one output neuron are present.
**Step 4:** The weights are added to the designed network and also it is biased. The developed Network is shown in Figure 5.8.

**Step 5:** The basis function and the activation function which are chosen for the designed network are given below:

\[ Z_i = \alpha + \sum_{j=1}^{N_z} w_{ij} \phi\left( \left| r^{(R)}_ij - v_{ij} \right| \right) \tag{5.7} \]

\[ \phi\left( \left| r^{(R)}_ij - v_{ij} \right| \right) = \exp\left[ -\gamma \left( r^{(R)}_ij - v_{ij} \right)^2 \right] \quad \text{(Sathish 2004)} \tag{5.8} \]

Here, \( P^{(R)}_i \) is the vector of \( r^{(R)}_i, g^{(R)}_i \) and \( b^{(R)}_i \) which represents the \( i^{th} \) pixel of the \( R^{th} \) region which is the regions generated from the river database images. Here \( v_{ij} \) is the center vector for the neuron. \( \alpha \) and \( w_{ij} \) is the learning rate and weight assigned for the neuron. Equations (5.7) and (5.8) are the functions for the designed neural network Net. The output of the network \( N \) is obtained by giving the region vector as its input.

**Step 5:** The learning error is determined for the network Net as follows:

\[ Er = \frac{1}{N_h} \sum_{a=0}^{N_z} D_a - Z_o \tag{5.9} \]

Here, \( Er \) is the error in the designed RBFN network; \( D_0 \) is the desired output and \( Z_0 \) is the actual output.
5.4.2 Minimization of Error by Bayesian Regulation Back Propagation Algorithm

The steps involved in the training of BRBP algorithm based Net is given below:

\[ j = jm \times jm \]  
\( (5.10) \)

\[ e = jm \times er \]  
\( (5.11) \)

\[ dx = -(j + I * mu) / e \]  
\( (5.12) \)

This network training function updates the weight and bias values according to Levenberg-Marquardt optimization in Matlab. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. The process is called Bayesian regularization.

1. Randomly generate weights in the interval \([0,1]\) are assigned to the neurons of the hidden layer and the output layer.

2. Determine the error using Equation (5.9), and give the training data sequence as input to the NN vide Equations (5.7) and Equation (5.8)

3. Adjust the weights of all the neurons when the BRBP error is determined as follows,

\[ w_{ij} = w_{ij} + \Delta w_{ij} \]  
\( (5.13) \)

The change in weight \( \Delta w_{ij} \) given in Equation (5.13) can be determined as \( \Delta w_{ij} = \bar{\sigma} . Z_{ij} . Er \), where \( Er \) is the BRBP error and \( \bar{\sigma} \) is the learning rate and it normally ranges from 0.2 to 0.6.
(4) After adjusting the weights, repeat steps (6) and (7) until the BP error gets minimized.

Normally, it is repeated till the criterion, $Er \leq 0.1$ is satisfied.

Once the error gets minimized to a minimum value it is construed that the designed RBFN network is trained for its further testing phase and the BRBP algorithm is terminated. Thus the neural network is trained using the regions of the river database images. Now the trained network for detecting the watered regions of the satellite river image is obtained. These identified regions are utilized for the further process.

### 5.4.3 Triangulation and Length Computation

Subsequent to the process of identifying the watered regions in the satellite river images, these identified regions are subjected to the triangulation process to form the triangle mesh. In computer graphics, a triangle mesh is a kind of a polygon mesh. It consists of a set of triangles joined by their common edges or corners (normally in three dimensions). All the identified water regions are subjected to triangulation so that the regions are filled up with networked triangles. Thus the triangle mesh $Tr$ and the length $L_{Tr}$ of the triangle mesh are calculated. This length is very important which is utilized to analyze the stage of the river. This obtained length is utilized to predict the stage of the river. The training phase which is utilized to predict the stage of the river is detailed below.
5.4.4 Training Phase to Predict the Stage of River

The proposed technique is comprised of one more training process which are utilized to identify the changes in the water level. In order to identify the stage of the river from the obtained satellite images, the RBFN network with BRBP algorithm is utilized to scrutinize the stage of the river by means of the length which is computed. The RBFN network is utilized to compute the stage of the river water based on some thresholds and this network is utilized to train the status weight of the identified regions. The length of different regions is given as the input and the RBFN network may decide the stage of the river based on this length. The trained single network NN accepts a collection of different length as input and gives a corresponding weight as its output. Hence the network is configured with single input and single output. Let \( L \) be the vector of different lengths and \( l_{\text{min}} \) and \( l_{\text{max}} \), be the minimum and maximum values. The following steps are used to identify the water level:

**Step 1:** As the first step, design a three layered neural network with one input layer one hidden layer and a single output layer. The input layer must have a single neuron to accept the different length input vector \( L \), and the output layer also must have a single neuron to interpret the change in water level and provide the status weight as the output. The hidden layer must have \( N_{\text{hd}} \) numbers of hidden neurons and a bias neuron.

**Step 2:** Set up the input weights and bias for the designed \( NN \). Assign weights only to those neurons that are present in the hidden and output layer. Figure 5.9 informs the outcome of the designed RBFN network.
Figure 5.9 Designed RBFN Network to Identify the Stage of River Water

**Step 3:** The basis function and the activation function which are chosen for the designed NN are given below.

\[ Y_n = \tau + \sum_{m=0}^{N_{hd}-1} w_{mn} \phi \left( \| x_n - u_{mn} \| \right) \quad (5.14) \]

\[ \phi \left( \| x_n - u_{mn} \| \right) = \exp \left[ -\eta \| x_n - u_{mn} \|^2 \right] \quad (5.15) \]

Here Equations (5.14) and (5.15) are the activation and the basis functions and \( u_{mn} \) is the center vector for the neuron. \( \tau, \eta \) are the bias and \( w_{mn} \) is the weight assigned for the neuron.

**Step 4:** The learning error is determined for the NN as follows:

\[ Err = \frac{1}{N_{hd}} \sum_{x=0}^{N_{hd}-1} D_x - Y_x \quad (5.16) \]

Here, \( Err \) is the error in the designed RBFN network NN; \( D_x \) is the desired output and \( Y_x \) is the actual output.
5.4.5 Status of Water Level Identification

The length vector $L_{Tr}$ of triangle mesh is then utilized to predict the stage of the river. This is achieved through the designed RBFN network NN described in section 5.3.1 by giving the length vector $L_{Tr}$ as its input. It outputs appropriate status weight for each length present in the length vector. The resulting images shown in Figures 5.10.

![Figure 5.10 Stage of Input River](image)

5.5 TESTING PHASE

In this testing phase, an input satellite river image is given to the preprocessing phase. In this phase, the input data is given as input to the preprocessing phase (section 5.1). As an outcome of the preprocessing phase, the region identified images have been obtained. The identified regions are given as input to the designed RBFN network Net, for identifying the watered area of the regions. After identifying the watered area of the image, the triangle mesh of the watered region is identified. Then the length of the watered region is identified and it is given as input to the designed network NN. The ranges defined in this work are 0-0.3 for drought, 0.4-0.6 for normal and 0.8 to 1.0 for flooding. In the network based on the length the stage of the river has been identified. Thus, the stage of the input river is obtained from its image. The output of the radial basis feed forward network by varying the number of neurons, number of hidden layers and learning rate is provided in
Tables 5.1 and 5.2 and the corresponding screen shots are shown in Figures 5.11 and 5.12.

**Table 5.1 MSE by Varying the Learning Rate for RBFNN**

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Learning Rate</th>
<th>No. of Hidden Layers</th>
<th>No. of Neurons in the hidden layer</th>
<th>Plot MSE</th>
<th>Performance MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>3</td>
<td>0.2</td>
<td>1</td>
<td>50</td>
<td>0.030065</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td>0.0205843</td>
<td>0.02017</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>150</td>
<td>0.0102687</td>
<td>0.03851</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>200</td>
<td>0.00345319</td>
<td>0.02854</td>
</tr>
<tr>
<td>2.</td>
<td>0.4</td>
<td>1</td>
<td>50</td>
<td>0.024617</td>
<td>0.01478</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td>0.03548</td>
<td>0.02874</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>150</td>
<td>0.01458</td>
<td>0.01874</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>200</td>
<td>0.01234</td>
<td>0.04254</td>
</tr>
<tr>
<td>3.</td>
<td>0.6</td>
<td>1</td>
<td>50</td>
<td>0.046417</td>
<td>0.03547</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td>0.021708</td>
<td>0.02154</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>150</td>
<td>0.011391</td>
<td>0.01057</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>200</td>
<td>0.0032951</td>
<td>0.01547</td>
</tr>
</tbody>
</table>

**Table 5.2 MSE by Varying Hidden Layers and Hidden Neurons for RBFNN**

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>No. of Hidden Layers</th>
<th>No. of Neurons in the hidden layer</th>
<th>Learning Rate</th>
<th>Plot MSE</th>
<th>Performance MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1</td>
<td>50</td>
<td>0.2</td>
<td>0.0210</td>
<td>0.03145</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>50,1</td>
<td>0.2</td>
<td>0.0310</td>
<td>0.0417</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>50,1,1</td>
<td>0.2</td>
<td>0.0417</td>
<td>0.0489</td>
</tr>
<tr>
<td>2.</td>
<td>1</td>
<td>100</td>
<td>0.2</td>
<td>0.0126</td>
<td>0.0138</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>100,1</td>
<td>0.2</td>
<td>0.0478</td>
<td>0.0518</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>100,1,1</td>
<td>0.2</td>
<td>0.0371</td>
<td>0.0670</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>100,1,1,1</td>
<td>0.2</td>
<td>0.0571</td>
<td>0.0681</td>
</tr>
</tbody>
</table>
Figure 5.11 Neural Network Training-RBFNN

Figure 5.12 Neural Network Training – Performance-RBFNN
5.6 RESULTS AND DISCUSSION

The proposed mechanism to predict the river water level changes was implemented on the working platform of MATLAB (version 7.12). The proposed mechanism was evaluated through the river image database used to design the network and the input river image of the Vaigai River was analyzed. The stage of this river was obtained through the step by step process and the results attained by this mechanism are shown.

In the designed RBFN network for water region identification, a collection of images was utilized to train the network. For the other neural network that is designed to output the status weight, a sequence in the range of $l_{\text{min}} = 0$ to $l_{\text{max}} = 350$ was utilized to train the network. The input image utilized for the analysis phase was that of the Vaigai River and the image is initially segmented using nonlinear diffusion method and then converted into a binary image. All the preprocessing and analysis operations were carried out to identify the river water regions. For the given input river image, drought stage was identified as the water level stage. So, it is evident that the proposed mechanism shows satisfactory performance because the stage of the Vaigai river is identified correctly.
Table 5.3 Performance Analysis for RBFNN

<table>
<thead>
<tr>
<th>Stage of river</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Sensitivity</th>
<th>FPR</th>
<th>Accuracy</th>
<th>Specificity</th>
<th>PPV</th>
<th>NPV</th>
<th>FDR</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>50</td>
<td>25</td>
<td>66.66667</td>
<td>75</td>
<td>50</td>
<td>75</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>Draught</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>50</td>
<td>0</td>
<td>83.33333</td>
<td>100</td>
<td>100</td>
<td>80</td>
<td>0</td>
<td>63.24555</td>
</tr>
<tr>
<td>Flood</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>25</td>
<td>83.33333</td>
<td>75</td>
<td>66.66667</td>
<td>100</td>
<td>33.33333</td>
<td>64.54972</td>
</tr>
</tbody>
</table>
5.7 SUMMARY

In this work, an effective mechanism to predict the stage of water level from satellite images using Radial basis feed forward network is proposed. The proposed mechanism was trained with two neural networks one to identify the water regions and the other to determine the stage of the river from its water level. The raw input image of Vaigai river was de-noised and then it was segmented into different regions with the aid of the designed RBFNN. Then a morphological operation was carried out on the segmented image. After this, the stage of the river was analyzed through another neural network. Finally, the stage of the river was identified as ‘Drought’ stage. The results proved the proficiency of the proposed method in determining the status of the river from its satellite image.