# CHAPTER 6
# MODELLING METHODOLOGY

The dataset gathered through experimental investigations were used for development of predictive eutrophication models. For predicting eutrophication indicators DO, SD and Chl-a, popular data-driven technique ANN was used. Under ANN topology, most commonly employed ANN architecture i.e. multilayer perceptron (MLP) is used as discussed in Chapter 2. To compare the performance of static MLP network with other ANN architectures a dynamic time delay neural network (TDNN) was also explored for model development. The neural network models were further compared with non-parametric machine learning models based on support vector regression (SVR) and Gaussian process regression (GPR) techniques which are explored less in lake eutrophication modelling. Moreover, prediction accuracy of these approaches were further compared with sophisticated Adaptive Neuro-Fuzzy Inference System (ANFIS) which generally performs well for regression and classification tasks as discussed in Chapter 2.

Based on the dataset from the two experimental investigations conducted, two sets of models have been trained using the machine learning algorithms. Initially the input parameters were chosen out of different investigated water quality parameters using a model-based approach in ANN architecture. Thereafter the same inputs were used in other data-driven learning approaches for development of the predictive eutrophication models respectively. The topology of ANN, MLR, GPR, SVR, and ANFIS models developed are presented in the next sections. The models were trained, tested, and validated in MATLAB environment and post processing works were done in Microsoft Excel.

## 6.1 INPUT AND OUTPUT PARAMETER SELECTION

DO, SD, and Chl-a are the most commonly used eutrophication indicators that efficiently describe the ecosystem health, water quality, and algal load on surface water bodies. The present study uses DO, SD, and Chl-a as model output parameter for

eutrophication assessment. For development of the predicting models, proper input parameters selection is one of the most crucial tasks for better performance. However, most of the earlier ecological data-driven models developed, lesser attention has been given to this aspect. In most of the cases inputs were chosen based on a model free approach such as domain knowledge or ad-hoc basis, which may result in too many or too few inputs [112]. Therefore, in the present study a stepwise model-based data pruning approach as suggested by Maier et. al. was used for selection of best combination of input parameters for the proposed models [112, 193]. Different parameters such as pH, total dissolved solids (TDS), electrical conductivity (EC), biochemical oxygen demand (BOD), turbidity, total nitrogen (TN), total phosphorus (TP), chlorophyll-a (Chl-a) and water temperature were chosen for optimization to be used as model input parameters. Different combinations of inputs were tested under ANN architecture for each target prediction. Once the input parameters were finalized in ANN for DO, SD and Chl-a prediction, the same parameters were used in MLR, SVR, GPR, and ANFIS models for target prediction respectively.

## 6.2 MLR MODEL

A linear regression model is used to describe the relationship between an output variable (y) and an input variable (x). The output variable in a linear regression model is written as an equation that is linear in the regression coefficient of the input variable. The output variable in a multiple linear regression model is dependent on more than one input variable and is stated as the sum of a constant term and additional terms. A general MLR model of the form shown in Equation (6.1) is used in this study for the desired output parameters DO, SD, and Chl-a.

$$y_i = \alpha_0 + \alpha_1 x_{i1} + \alpha_2 x_{i2} + \cdots + \alpha_p x_{ip} + \epsilon_i, \quad i = 1, \ldots, n \tag{6.1}$$

Where, n is the number of observations.

$y_i$ is the $i^{th}$ response for DO, SD, or Chl-a.

$\alpha_p$ is the $p^{th}$ coefficient and $\alpha_0$ is the constant term.

$x_{ij}$ is the $i^{th}$ observation of $j^{th}$ input variable.

$\varepsilon_i$ is the $i^{th}$ random error.

The MLR model was developed in the statistical and machine learning toolbox of MATLAB and the results were evaluated based on statistical indices.

## 6.3 ANN MODEL

Artificial neural networks are very useful computational technique for modelling complex non-linear systems particularly when the underlying data relations are quite involuted like in case of ecological systems [172]. ANN is a data processing system which performs in a similar way in which a human brain performs a specific task of interest [64]. In ANN topology, an output is calculated based on three unique components which are weights (w), bias (b) and an activation function (f) as given in Equation (6.2) below.

$$y = f(w.x + b) \tag{6.2}$$

where, y and x are the output and input parameters respectively. Figure 6.1 shows the architecture of the ANN models developed for prediction of DO, SD, and Chl-a in eutrophic lakes. For each model types developed, three layers were considered having only one hidden layer. As single variable forecasting was targeted, only one neuron in the output layer was extensively used while number of neurons in the input and hidden layer were heuristically determined.

### 6.3.1 ANN topology

ANN models were developed with neural network toolbox in MATLAB (release 2017a). A feedforward neural network was used with backpropagation learning algorithm which were generally used successfully to model non-linear complex systems [13, 92, 95]. In this study, two types of ANN models were used for target prediction in eutrophic lakes of Assam. First one is a static type of multilayer perceptron (MLP) and second one is dynamic type time delay neural network (TDNN). Basic difference between the two types is that time is not a consideration for prediction in the static models, while effect of earlier data is also considered for future data prediction in the dynamic models [11]. The TDNN used in this study was a non-linear autoregressive network with exogenous inputs (NARX) which is a recurrent dynamic network [37].

The Levenberg-Marquardt backpropagation training algorithm was used in this study. It is an iterative technique for finding the local minimum of a multivariate function written as the sum of squares of many non-linear real valued functions. It interpolates between the Gauss-Newton technique and the gradient descent approach [105]. The Levenberg-Marquardt backpropagation training algorithm is having faster convergence compared to other methods and is highly recommended [37]. To introduce non-linearity in the models, a sigmoid tangent activation function (TANSIG) was used in hidden layer which produces output in the range +1 to -1. A linear transfer function (PURELIN) was used in the output layer. The hyperbolic tangent sigmoid function is given in Equation (6.3) and the linear function is given in Equation (6.4), where f(x) represents the output for dependent variable x.

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \tag{6.3}$$

$$f(x) = x \tag{6.4}$$

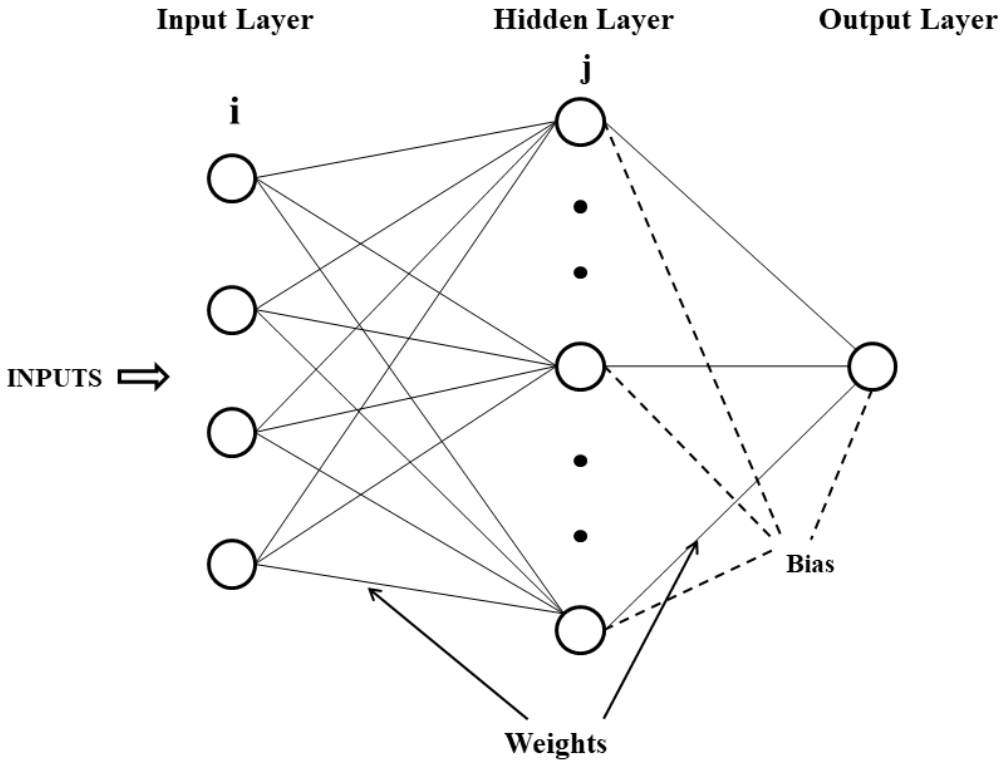


**Figure 6.1:** ANN architecture for the developed models

### 6.3.2 Fixing number of neurons in hidden layer

A single hidden layer was considered to avoid complexity of all the models developed. Optimum number of neuron selection in the hidden layer is very crucial for better generalization and to prevent over fitting of the model. As there is no hard and fast rule available in literature for obtaining the optimum number of neurons in the hidden layers, so a trial and error method based on previous literature was used in this study. Table 6.1 below presents some of such earlier methodologies for fixing optimum number of neurons in the hidden layer of a neural network. The five empirical methods cited were used to calculate the number of neurons required for the hidden layer in a model and thereafter several neural networks were created between the minimum and maximum values. Based on its mean squared error (MSE) and correlation coefficient (R) between the output and target values, each network was evaluated, and optimum number of neurons was finalized.

**Table 6.1:** Previous approaches for fixing number of hidden layer neurons

| Sl. No. | Method | References |
|---|---|---|
| 1 | $N = \frac{4n^2 + 3}{n^2 - 8}$ | [167] |
| 2 | $N = \frac{\sqrt{1 + 8n} - 1}{2}$ | [99] |
| 3 | $N = \sqrt{N_i N_o}$ | [168] |
| 4 | $N = \frac{N_i + \sqrt{N_p}}{L}$ | [80] |
| 5 | $N = \frac{2^n}{n} + 1$ | [207] |

Where, **N** = No of neurons in hidden layer; $\mathbf{N_i}$ & **n** = No of input neurons

$\mathbf{N_o}$ = No of output neurons; $\mathbf{N_p}$= No of input sample; **L** = No of hidden layers

### 6.3.3 Data normalization

Data normalization is very essential to minimize bias within the dataset or to eliminate variable dimensionality. A general linear transformation as given in equation (9) below was used for data normalization [71]. Using the Equation (6.5), the whole dataset was transformed into specified interval of 0.15 to 0.85.

$$x_i' = lower + \frac{(x_i - x_{min}) \times (upper - lower)}{(x_{max} - x_{min})} \quad (6.5)$$

where, $x_i'$ and $x_i$ are the standardised and actual data respectively. $x_{max}$ and $x_{min}$ are the maximum and minimum values of $x_i$ and upper and lower corresponds to the data normalization range. The whole database was divided into three parts randomly into training, testing and validation subsets after normalization. 70 percent data were used for training the model and remaining 30% of data were used in equal parts for model testing and validation respectively. After completion of model training and testing, the model outputs were denormalized to get the real data by reversing the action.

## 6.4 SVR ARCHITECTURE

Support vector machines (SVM), developed by Vapnik [183], are supervised machine learning methods used mainly in the domain of classification and regression problems. The basic aim of SVM algorithm is to locate the best hyperplane by converting the original input space ($x_i$) into a higher dimensional feature space through a non-linear mapping function ($\phi(x)$) [120]. By introducing a ε-insensitive loss function, SVM is used for solving regression problems and are popularly termed as SVR [54]. ε-insensitive loss function for the target variable (y) and predictor variable (x) can be defined as given in Equation (6.6).

$$L(y) = 0 \; for \; |f(x) - y| < \varepsilon; \; otherwise \; L(y) = \; |f(x) - y| - \varepsilon \quad (6.6)$$

The objective function f(x) in SVR can be expressed in the following form (Equation (6.7)).

$$y = f(x) = \{w.\emptyset(x)\} + b \quad (6.7)$$

where w and b are the vector coefficient and a constant respectively. Values of these parameters are evaluated by minimizing a regularized risk function with two slack variables as given in Equation (6.8) and associated boundary conditions given in Equation (6.9).

$$minimize \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{6.8}$$

$$Subjected\ to \begin{cases} y_i - w.\emptyset(x_i + b) \leq \varepsilon + \xi_i \\ w.\emptyset(x_i + b) - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \quad (i = 1,2,3, \ldots . n) \end{cases} \tag{6.9}$$

Here $\xi_i$ and $\xi_i^*$ are the slack variables introduced to handle infeasible constraints. C is a constant that decide trade-off between model flatness and training error and i represents n number of training samples. Finally with the help of kernel trick, the regression formula (Equation (6.7)) can be converted into non-linear SVR as given by dual formula in Equation (6.10) for prediction of new values.

$$f(x) = \sum_{i-1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x_j) + b \tag{6.10}$$

where $\alpha_i$ and $\alpha i^*$ are the Lagrange multipliers and kernel function is represented by $K(x_i, x_j)$.

The kernel function is used for non-linear data transformation and hence selection of appropriate kernel parameter is very important for satisfactory model performance [139]. The common kernel functions can be represented with Equations (6.11) to (6.13) as below:

Linear: $K(x_i, x_j) = x_i' x_j$ (6.11)

Gaussian: $K(x_i, x_j) = e^{\left(-\|x_i - x_j\|^2\right)}$ (6.12)

Polynomial: $K(x_i, x_j) = (1 + x_i' x_j)^q,\ where\ q\ is\ in\ the\ set\ (2.3, \ldots)$ (6.13)

So, in this study four commonly used kernel functions viz. linear, quadratic, cubic and Gaussian kernels were compared under SVR method for DO, SD, and Chl-a prediction. Based on the results of coefficient of determination ($R^2$) and root mean square error (RMSE), best kernel was chosen for each model. In SVR models, C and ε are very important parameters that vary depending upon the noise associated with input data. These parameters are unknown for best model performance and so a trial and error

method was used to find the best combination of C and ε for DO, SD, and Chl-a prediction.

## 6.5 GPR ARCHITECTURE

GPR is a type of non-parametric machine learning modelling approach that trains sample data using probabilistic approaches that accounts for uncertainty about the target variable [101, 155]. GPR technique in general is very robust and precise compared to other regression methods [170]. In GPR algorithm the targeted variable is considered to be under multivariate normal distribution. A mean function and a covariance or kernel function is used to specify the Gaussian process [143] and posterior probability hypothesis is ascertained using Bayesian inference method. For a target variable (y(k)), a Gaussian process can be expressed in terms of a random variable (x(k)) by the following expression as given in Equation (6.14).

$$y(k) = f[x(k)] + \zeta(k) \tag{6.14}$$

where $\zeta(k) \sim N(0, \sigma^2)$ is Gaussian noise having variance $\sigma^2$. Here it is considered that error $\zeta$ is under normal distribution and has zero mean variance $\sigma^2$. The function f(x) is driven by a Gaussian process on x and it is specified by a kernel such that,

$$y(x) = (x_1, \ldots, x_N) \sim N(0, K + \sigma^2 I) \tag{6.15}$$

where K is the kernel or covariance matrix and I is the identity matrix. After fixing the Gaussian noise, a Bayesian inference is applied that minimizes the negative log-posterior (Equation (6.16)) to train the model under GPR [38].

$$p(\sigma^2, k) = \frac{1}{2} y^T (K + \sigma^2 I)^{-1} y + \frac{1}{2} \log|K + \sigma^2 I| - \log p(\sigma^2) - \log p(k) \tag{6.16}$$

As Gaussian process can be fully defined by its second order statistics, a process with zero mean implies that the covariance function will completely determine the behaviour of the process [52]. Hence choice of proper covariance function becomes

necessary. There are several types of covariance functions commonly used for GPR such as squared exponential, matern 5/2, matern 3/2, exponential, rational quadratic, polynomial etc. [144]. The covariance function k ($x_i$, $x_j$ | θ) for predictor variables $x_i$ and $x_j$, parameterized in terms of vector θ can be defined as follows:

Squared exponential kernel: $$k(x_i, x_j|\theta) = \sigma_f^2 e^{\left[-\frac{1}{2}\frac{(x_i-x_j)^T(x_i-x_j)}{\sigma_l^2}\right]} \quad (6.17)$$

Exponential Kernel: $$k(x_i, x_j|\theta) = \sigma_f^2 e^{\left(-\frac{r}{\sigma_l}\right)} \quad (6.18)$$

Rational quadratic kernel: $$k(x_i, x_j|\theta) = \sigma_f^2 \left(1 + \frac{r^2}{2\alpha\sigma_l^2}\right)^{-\alpha} \quad (6.19)$$

Matern 5/2 kernel: $$k(x_i, x_j|\theta) = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2}\right) exp\left(-\frac{\sqrt{5}r}{\sigma_l}\right) \quad (6.20)$$

Here $\sigma_l$ represents characteristic length scale and $\sigma_f$ is the standard deviation. r is the Euclidean distance between $x_i$ and $x_j$.

In the presented work DO, SD, and Chl-a models were developed initially using four different types of GPR algorithms namely squared exponential, rational quadratic, matern 5/2 and exponential. Thereafter best GPR models were selected for each target variables and the same were utilized for comparison with the corresponding ANN and SVR based models.

## 6.6 ANFIS ARCHITECTURE

ANFIS is an adaptive network with neural learning capabilities is based on first order Sugeno fuzzy model and was introduced by Jang [73]. ANFIS creates a fuzzy inference system (FIS) based on given a input-output combination, whose membership function parameters are estimated using either back propagation or a hybrid rule [174]. Hybrid algorithms which combine a back propagation algorithm and a least square method can improve the learning efficiency and also the algorithm becomes simpler [87]. ANFIS is basically a graphical representation of Sugeno-type fuzzy system which is capable of constructing a network realization of fuzzy IF/THEN rules coupling advantages of both neural networks and fuzzy logic in a common framework [116]. For

example, the fuzzy rules for a single output (f) based on two inputs ($x_1$ and $x_2$) can be written in the following form.

Rule 1: If $x_1$ is $A_1$ and $x_2$ is $B_1$, then $f_1 = p_1 x_1 + q_1 x_2 + r_1$ (6.21)

Rule 1: If $x_1$ is $A_1$ and $x_2$ is $B_1$, then $f_2 = p_2 x_1 + q_2 x_2 + r_2$ (6.22)

where $A_i$ and $B_i$ are the fuzzy sets and $p_i$, $q_i$, and $r_i$ are the design parameters of the system.

The structure of ANFIS is a five-layer neural network, each having to perform specific fuzzy inference steps as shown in Figure 6.2 [3, 73].

*1st Layer:* Initial fuzzification layer which determines the membership functions associated with input values. In this layer the output or the degree of involvement ($O_1$) for input variable $x_1$ can be estimated by using the Equation (6.23).

$$O_{1,i} = \mu_{A_i}(x_1), \quad i = 1,2 \tag{6.23}$$

where $x_1$ is the input to node i; $A_i$ is the linguistic label and $\mu A_i$ is the membership function.

*2nd Layer*: Database construction layer where each node provides the strength of a rule. The output of this layer called as firing strength ($O_{2,i}$) is determined by product of all incoming signals from layer 1.

$$O_{2,i} = w_i = \mu_{A_i}(x_1) * \mu_{B_i}(x_1), \quad i = 1,2 \tag{6.24}$$

*3rd Layer:* Normalization layer which produces normalized firing strengths by dividing individual strength to total firing strength of all rules. Output in this layer is determined by Equation (6.25).

$$O_{3,i} = \overline{W_1} = \frac{w_i}{W_1 + W_2}, \quad i = 1,2 \tag{6.25}$$

*4th Layer:* Decision making layer where every node is adaptive node. Each node in this layer takes normalized inputs from 3rd layer and computes a linear function where the function coefficients are adapted by using error function of the feed-forward neural network. The output of this layer can be defined by the following Equation (6.26).

$$O_{4,i} = \overline{W_i}.f_1 = \overline{W_i}.(p_i.x + q_i.y + r_i) \tag{6.26}$$

Here $p_i$, $q_i$, and $r_i$ are the coefficients of the linear relationship and commonly called as consequent parameters.

*5th Layer:* Output layer where defuzzificated values are passed from 4th layer to this layer to obtain a single valued output as given by Equation (6.27).

$$Overall\ Output = O_{5,i} = \sum \overline{W_i}\,.f_i = \frac{\sum_i W_i.f_i}{\sum_i W_i} \tag{6.27}$$

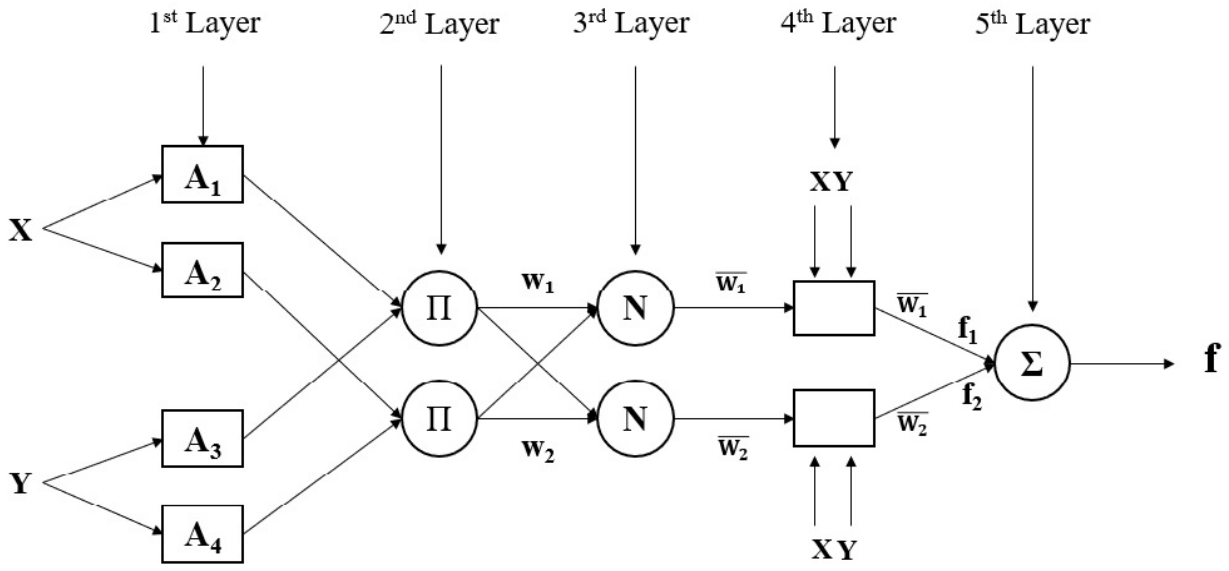


**Figure 6.2:** ANFIS structure for the developed models

ANFIS models were used in MATLAB (release 2017a) environment for prediction of DO, SD, and Chl-a in eutrophic lakes with laboratory investigated water quality parameters. In this work, subtractive fuzzy clustering method was used to automatically generate tuned membership functions. Under subtractive clustering, different combinations of range of influence and squash factor values were considered,

and each FIS systems were trained using a hybrid optimization method. After testing all the parameter combinations with statistical measures that describe the performance of the model, the optimum architecture of the proposed ANFIS models were finalized for further investigation.

## 6.7 MODEL EFFICIENCY EVALUATION

The efficiency of the developed models were analysed with the help of statistical parameters coefficient of determination ($R^2$), Nash-Sutcliffe efficiency (E) [125], root mean square error (RMSE), and mean absolute error (MAE).

### 6.7.1 Coefficient of determination ($R^2$)

The coefficient of determination ($R^2$) is a statistical measure that quantifies the amount of variance in the dependent variable that can be explained by the independent variables in a regression model. It's value ranges in between 0 and 1, with higher values suggesting a better fit of the model to the observed data. The goodness-of-fit of a model can be estimated with $R^2$ value. It can be calculated based on the formula given in Equation (6.28).

$$R^2 = \{\frac{n(\sum O_i.P_i)-(\sum O_i)(\sum P_i)}{\sqrt{[n.\sum O_i^2-(\sum P_i)^2][n.\sum O_i^2-(\sum P_i)^2]}}\}^2 \quad (6.28)$$

where, $O_i$ and $P_i$ are the experimentally observed and predicted values of the model respectively; $O_{mean}$ represents the arithmetic mean value of the observed quantity and n represents the number of data samples.

### 6.7.2 Nash-Sutcliffe efficiency (E)

Nash-Sutcliffe efficiency is another popular statistical index used for assessment of predictive capacity of ecological and water quality models. It provides a thorough evaluation of model performance by accounting for both the bias and variability of the model's predictions. E value ranges from negative infinity to 1, with 1 denoting an ideal

model fit, 0 denoting a poor model fit, and a negative value infers that the mean observed value is a better predictor than the model. E can be evaluated from the following equation.

$$E = 1 - \left(\frac{\sum(O_i - P_i)^2}{\sum(O_i - O_{mean})^2}\right) \tag{6.29}$$

### 6.7.3 Root mean square error (RMSE)

The RMSE of a regression or prediction model is a measure of the average magnitude of the residuals. It measures the square root of the average of the squared differences between observed and predicted values. RMSE accounts for both bias and variability in model prediction and provides a measure of goodness-of-fit of a model. RMSE is calculated based on the Equation (6.30) given below.

$$RMSE = \left[\frac{1}{n}(\sum|O_i - P_i|^2)\right]^{1/2} \tag{6.30}$$

### 6.7.4 Mean absolute error (MAE)

The average absolute difference between expected and actual values is measured by MAE. It is calculated by averaging the absolute values of the residuals as given in Equation (6.31). Due to its simplicity and ease of interpretation, MAE is widely used to evaluate performance of regression models.

$$MAE = \frac{1}{n}\sum|O_i - P_i| \tag{6.31}$$

## 6.8 MODEL VALIDATION

Eutrophication models for indicators DO, SD, and Chl-a were developed with the dataset of water quality parameters monitored on artificially simulated prototype lakes. Feasibility of the adopted modelling approach to be used as eutrophication predictor in natural waterbodies was evaluated by checking the model performance with samples collected from a few natural shallow waterbodies in Assam. The models'

prediction should be flawless under different ecological conditions and hence a vast dataset was gathered to check accuracy of the presented models to be used in waterbodies in Assam. Water samples were gathered from two sampling locations of Deepor Bil, a world heritage RAMSAR wetland in Guwahati city as well as from a marsh (two locations), a manmade lake, and a village pond in and around Tezpur University campus in Tezpur city, details of which are presented in Table 6.2 below. A total of 25 samples were collected from the waterbodies during different weather conditions in the month of March, May, September, and December 2019 and all the previously mentioned water quality parameters were investigated on the samples.

## 6.9 SENSITIVITY ANALYSIS

Sensitivity analysis is essential in data-driven models because it helps in the identification of influential variables, the understanding of model behaviour, the assessment of model robustness, decision-making and risk assessments. To determine the effect of each independent variable on the prediction of dependent variables, sensitivity analysis was done on the pre-trained DO, SD, and Chl-a models. After training and validation of the eutrophication models, comparison among adopted data-driven approaches was done and the best performing model types were used for sensitivity analysis. For sensitivity analysis, method based on data perturbation technique was used [19]. In the data perturbation method of sensitivity analysis, each input parameters were increased and decreased in succession, keeping all other input parameters unchanged. For the present study, each parameter were altered by ±20% and then sensitivity of individual parameters were calculated as percentage change in output parameters caused by percentage change in input parameters [113]. Parameters having higher sensitivity values were identified and relative importance of inputs on target prediction was identified.

**Table 6.2**: Details of water sample collection points for model testing

| Sampling point | Type of water body | Location | Latitude | Longitude |
|---|---|---|---|---|
| 1 | Deepor Bil (RAMSAR wetland) | Near Pamohi, Guwahati | 26º6'47.72" N | 91º39'35.76" E |
| 2 | Deepor Bil (RAMSAR wetland) | Near Pamohi, Guwahati | 26º6'46.32" N | 91º39'13.51" E |
| 3 | Marsh | Near Tezpur University, Tezpur | 26º41'13.97" N | 92º48'58.78" E |
| 4 | Marsh | Near Tezpur University, Tezpur | 26º41'16.37" N | 92º48'58.00" E |
| 5 | Village Pond | Near Tezpur University, Tezpur | 26º41'30.56" N | 92º49'15.22" E |
| 6 | Artificial Lake | Tezpur University, Tezpur | 26º42'3.78" N | 92º49'52.08" E |

## 6.10 SUMMARY

Based on the dataset gathered from the two trials of experimental investigation, data-driven modelling approach such as ANN, MLR, SVR, GPR, and ANFIS were used to train models for common eutrophication indicators DO, SD, and Chl-a. A model-based approach was used for fixing the optimum inputs for each target prediction in ANN architecture. Thereafter the same inputs were used in other data-driven learning approaches for development of the predictive eutrophication models respectively. The structure of the adopted modelling approaches have been summarized with Table 6.3. Performances of the models were evaluated with statistical parameters –$R^2$, E, RMSE, and MAE. The trained models were subsequently tested with natural water body data from samples collected from different waterbodies in Assam, India. Sensitivity analysis was performed on the best performing models using a data perturbation technique.

**Table 6.3**: Details of different model structure

| Sl. No. | Model Type | Particulars | Specifications |
|---|---|---|---|
| 1 | ANN | ANN Type | Multilayer perceptron (MLP) & Time delay neural network (TDNN) |
| 2 | | Network type | Feed Forward Backpropagation |
| 3 | | Training algorithm | Levenberg-Marquardt backpropagation (TRAINLM) |
| 4 | | Adaption learning function | Gradient Descent with Momentum Weight and Bias (LEARNGDM) |
| 5 | | Performance function | Mean Square Error (MSE) |
| 6 | | Transfer function | Sigmoid tangent activation function (TANSIG) |
| 7 | | Data division | Random (Dividerand) |
| 8 | | Number of epoch (Learning cycle) | 1000 iterations |
| 1 | SVR | Kernel function tested | Linear, quadratic, cubic and Gaussian kernel |
| 2 | | Performance function | $R^2$ and RMSE |
| 3 | | Cross-validation technique used | 5-fold cross-validation |
| 4 | | Optimizing parameters | Box constant (C) and ε-insensitive loss function |
| 1 | GPR | Covariance function tested | Squared exponential, matern 5/2, matern 3/2, exponential, rational quadratic, polynomial |
| 2 | | Performance function | $R^2$ and RMSE |
| 3 | | Cross-validation technique used | 5-fold cross-validation |
| 4 | | Basis function | Constant |
| 1 | ANFIS | Type of ANFIS | First order Sugeno fuzzy model |
| 2 | | Membership function | Subtractive fuzzy clustering |
| 3 | | Training method | Hybrid optimization method |
| 4 | | Data division | Random (4:1) |
| 5 | | Optimizing parameters | Range of influence, Squash factor, Accept ratio, Reject ratio, number of epoch |