CHAPTER 4

MATERIALS AND METHODS

4.1 INTRODUCTION

In this study, Nagapattinam district Sardine and Mackerel monthly landing prediction for the year 2010 has been investigated with 112 multivariate Neural Network Models (NNM), based on their efficiency to provide an accurate fits and forecasts on the monthly commercial landings. Recently, Neural Network techniques have been applied to predict the nonlinear Time Series (TS) in fish catch analysis; apart from their major applications in financial predictions. In this study, the ability of the NNM were analyzed in two different patterns like Long Time Series (LTS) from the year 1998 to 2010 (13 years) and the Short Time Series (STS) from 2005 to 2010 (6 years). The fish catch data were targeted with three satellite derived ocean environmental parameters such as, Chlorophyll-a, Sea Surface Temperature and Photosynthetically Available Radiation in Artificial Neural Network (ANN) model. A Decision Support System (DSS) for fish catch and management of Sardine and Mackerel is developed for providing spatially distributed index based fish abundance. The DSS results are validated against present PFZ advisory and also with the insitu Sardine and Mackerel landings. The methodology is explained schematically by flow chart diagram as shown in Figure 4.1. The methodology is discussed in detail through the following chapters.
Figure 4.1 Flowchart showing the methodology
4.2 LONG TIME SERIES PREPARATION

4.2.1 Base Map (Bathymetry) Preparation

For the preparation of Time series data in terms of CHL, SST and PAR for Nagapattinam district fishing ground, a 1000m isobath is considered. The navigational chart (No.7706) issued by National Hydrographic Office, Government of India, in the scale of 1:3,500,000 is used for preparing the bathymetry. The detailed 1000m isobath bathymetry map for the study area is shown in Figure 4.2.
4.2.2 Fishery Data

For the preparation of LTS and STS data in terms of Oil Sardine (*sardinella longiceps*) and Indian Mackerel (*Rastrelliger kanagurtha*), monthly landing details were obtained from the Central Marine Fisheries Research Institute (CMFRI) Cochin, database. The period considered for LTS is from 1998 to 2010, out of which the CMFRI data pertains to the period 1998-2009. For validation purpose, fish catch data for the year 2010 from fish landing centres were collected. Standardization of commercial catches through Catch Per Unit Effort (CPUE) could provide a powerful method for estimating trends in the stock abundance. Unfortunately, there were many aspects of fishermen’s behaviour that will cause CPUE to be not proportional to abundance even on a very small spatial scale (Hilborn and Walters 1992). So, for this study, Sardine and Mackerel landing details are considered instead of CPUE for the preparation of fish landing LTS. The data were collected by qualified and well-trained technical staff of CMFRI by following stratified multi-stage random sampling technique, in which the Sardine and Mackerel landings were recorded by covering landing centres along the Nagapattinam coast. The catch data for the year 2010 (on monthly basis) were collected in entire Nagapattinam coastal area from all forty four landing centres physically with the help of Department of Fisheries, Officials. A team of fourteen local fishermen were engaged to cover the entire 44 fish landing centres for the collection of fish catch data as per questionnaire. This data is used for testing and validating NNM predicted fish catch. The questionnaire used for this study is included in Appendix 1 in this thesis.
4.2.3 Satellite Data

4.2.3.1 Introduction

Ocean colour remote sensing is applied more and more widely in the water quality monitoring of oceanic, Case 1 and Case 2 water bodies, because it has many advantages, such as, wide range synchronization and low cost for data collection. Satellite remote sensing of ocean colour information on chlorophyll concentration, SST, wind speed and Sea Surface Height (SSH) gives better understanding about Oceanographic processes such as, currents, ocean fronts, rings, eddies and coastal upwelling. In this context, the first Space borne ocean- colour sensor Coastal Zone Colour Sensor (CZCS) was launched in the year 1978, which provided data until 1986. This was followed by several new sensors such as, SeaWiFS (1997), MODIS in 1999 and 2002 and recently by OCEANSAT-2 OCM on Sep 23 2009, is added to the list. In this study, three satellite derived ocean environmental parameters (CHL), (SST) and (PAR) are considered as input variable for NNM. The different data products used in the study are presented in Table 4.1.

Table 4.1 Details of satellite data used in this study

<table>
<thead>
<tr>
<th>Sl. No</th>
<th>Description of Data Products</th>
<th>Spatial Resolution</th>
<th>Year</th>
<th>Number of images processed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Monthly L3 m CHL images (merged)</td>
<td>4.5 km</td>
<td>1998-2010</td>
<td>156</td>
</tr>
<tr>
<td>2.</td>
<td>Monthly L3 m CHL images (MODIS)</td>
<td>4.5 km</td>
<td>2005-2010</td>
<td>72</td>
</tr>
<tr>
<td>3.</td>
<td>Monthly L3 m SST images</td>
<td>4.5 km</td>
<td>1998-2010</td>
<td>156</td>
</tr>
<tr>
<td>4.</td>
<td>Monthly L3 m PAR images</td>
<td>9.0 km</td>
<td>1998-2010</td>
<td>156</td>
</tr>
<tr>
<td>5.</td>
<td>MODIS L1 A images</td>
<td>1.0 km</td>
<td>2010</td>
<td>03</td>
</tr>
<tr>
<td>6.</td>
<td>OCEAN SAT-2 OCM</td>
<td>360 m</td>
<td>2010</td>
<td>02</td>
</tr>
</tbody>
</table>
4.2.3.2 Chlorophyll-a Concentration (CHL)

The primary function of Chlorophyll is photosynthesis of marine algae present in the ocean. Algae is the main food for Sardine and Mackerel larvae and that availability of food during the critical developmental period of Sardine and Mackerel larvae determines the year class of Sardine & Mackerel population and hence CHL is one of the important indicator variable for Sardine and Mackerel availability in Bay of Bengal (BoB). So chlorophyll in Bay of Bengal is considered in NN prediction models for this study. SeaWiFS/MODIS/MERIS level 3 standard binned merged images archived by the Ocean Global colour website were used to estimate sea-surface case-1 CHL concentrations. These data were obtained freely in NetCDF format from the website ftp://ftp.acri.fr/GLOB_4KM/RAN/CHL1/MERGED/GSM. The images are Global Area Coverage (GAC) monthly composite SeaWiFS/ MODIS/ MERIS data with a spatial resolution of about 4.5 km for the period are considered from January 1998 to December 2010.

4.2.3.3 Sea Surface Temperature (SST)

In Sardine and Mackerel life cycle SST plays an important role, which leads their activity levels to increase or decrease. SST variation makes both the species move into certain areas, and influences their feeding and reproductive activity. Since the oil Sardine and Indian Mackerel are a tropical fishes, it is necesssary to understand how temperature affects their behavior. Hence SST is considered as one of the main environmental factor in the Sardine landing prediction, with preferred temperature range of 27° to 29°C (Chidambaram, 1950). The NOAA pathfinder data presented freely in ftp://podaac.jpl.nasa.gov/allData/avhrr/L3/pathfinder_v5/monthly/ were used to study SST with the same spatial resolution (4.5 km) like chlorophyll data.
The data available in the above website is up to 2009 only. So, for NNM testing purposes for the year 2010 (January - December) the data were extracted from the other website http://oceancolor.gsfc.nasa.gov/cgi/l3.

4.2.3.4 Photosynthetically Available Radiation (PAR)

PAR is the amount of light available for photosynthesis, which is defined as the quantum energy flux from the Sun light in the 400 to 700 nanometer wavelength range. The determination of PAR on the sea surface is the key issue, to estimate the primary productivity. The PAR at the surface for the day, is computed with remote sensing reflectance from SeaWiFS sensor data. Since, both the fishes are herbivores, PAR may also influences their migration and their feeding activity. Hence PAR also is considered as one of the biophysical parameter to design the NNM in this study. This data set consists of derived estimates of global PAR from Sea-viewing Wide Field-of-View Sensor (SeaWiFS), in orbit on the OrbView-2 (formerly Sea Star) platform. SeaWiFS data products are processed and distributed by the Ocean Biology Processing Group (OBPG). For this study, SeaWiFS Level 3 monthly Binned data files in HDF format were downloaded from the FTP site http://oceancolor.gsfc.nasa.gov/cgi/l3. SeaWiFS Level-3 binned monthly data product have 9Km spatial resolution, which is different from chlorophyll and SST spatial resolution.
Figure 4.3 Flowchart showing methodology for image DN value extraction
4.2.3.5  Image Processing

Monthly mean CHL, SST and PAR images for the entire period were downloaded, processed and extracted in BEAM software developed by European Space Agency (ESA). BEAM is an open-source toolbox and development platform for viewing, analyzing and processing of remote sensing raster data. BEAM supports a number of raster data formats such as GeoTIFF, HDF and NetCDF as well as data formats of other EO sensors such as Moderate Resolution Imaging Spectroradiometer (MODIS) and Advanced Very High Resolution Radiometer (AVHRR). The remote sensing data extracted for this study was in both NetCDF and HDF format, which is supported by BEAM software. The CHL and PAR products were clipped for the study area and the same is shown in Figure 4.4. A total of 540 images as listed in Table 4.1 were downloaded, processed and spatially averaged as point data for each polygon for this TS preparation. The procedure to create the 1000m isobath shape file for the study area is explained in detail in the flow chart shown as Figure 4.3.

In India previous studies on monthly CHL extraction have been performed with 200m isobath to get monthly averaged surface CHL for fisheries studies along the waters of the southwest coast of India George et al (2012). However because of multiday fishing activity and their ability to move form continental shelf area to continental slope area for fishing, a 1000m isobath has been considered in this study. Same 1000m isobath vector files were created for all coastal district of all Tamil nadu to extract monthly mean information for all the three oceanographic biophysical environmental parameters for the same period.
The extracted satellite derived environmental parameters CHL, SST & PAR values are spatially averaged for the preparation of long Time Series for the period 1998 to 2010 is depicted in Figure 4.4 and 4.5. The fishermen from Nagapattinam district and particularly those who use trawlers...
are involved in multiday fishing activity. They move out from the Nagapattinam district boundary up to Tiruvallur district in the northern side and Ramanathpuram district in the southern side (Questionnaire information). Hence, the spatially averaging of all three environmental variable values was created for all districts to derive monthly mean values. Sardine and Mackerel landing LTS of Nagapattinam district, from the year 1998 to 2009 are obtained from CMFRI are considered as input in this study. The year 2010 Sardine and Mackerel landing data are collected from the landing centres and placed as target in this study. The entire datasets of extracted CHL, SST and PAR and the Sardine and Mackerel landing details are available in the annexure II of this thesis for reference.

4.3 SHORT TIME SERIES PREPARATION

Apart from long time trends and seasonality, effective prediction may be accounted with short term impact on the study area. Gutie´rrez-Estrada et al (2008) reported that short-medium time period prediction/ forecast of a TS is necessary to frame the usefulness of any model. For STS preparation the TS period is from 2005 to 2010. The reason for considering 2005 is that, the study area Nagapattinam was heavily affected by tsunami in the year 2004. In STS NNM three types of CHL combinations were considered. The details of all three TS preparation were explained in detail in the next three sections 4.3.1 to 4.3.3.

4.3.1 Merged Chlorophyll-a Used for Short Time Series

NN analysis of all parameters followed in the LTS was considered in STS except for the length of TS. The TS duration considered in this STS is from the year 2005 to 2010. The number of samples involved in this Neural Network is 60-180 for training and validation in each parameter and whereas for testing phase it is 12 as in LTS. The remaining two environmental
parameters were the same as used in LTS from the year 2005. The long and STS orientation for the entire NNM is emphasized in the following flowchart depicted as Figure 4.6.

![Flowchart showing methodology in preparation of time series](image)

**Figure 4.6 Flowchart showing methodology in preparation of time series**

### 4.3.2 MODIS Chlorophyll-a used for Short Time Series

For this STS NNM the input variables SST and PAR were considered as the same as used in STS merged CHL explained in previous section 4.3.1. In merged STS, CHL is derived from merged products of SeaWiFS/MODIS/MERIS. But in this STS, MODIS CHL is obtained from
the website http://oceancolor.gsfc.nasa.gov/cgi/l3, which is available from the year 2002 having same spatial resolution. The extraction procedure for MODIS CHL preparation is as same as followed in the previous LTS preparation.

4.3.3 MODIS Validated Chlorophyll-a used for Short Time Series

In this STS except CHL all other parameters were same as the previous STS. The MODIS CHL was validated in BoB and compared with OCEANSAT-2 OCM products. For CHL validation two cruise trips were undertaken in BoB in the year 2010. A total of 71 geo-points water samples were collected and the samples were analyzed for validating with MODIS data for the preparation of MODIS Validated CHL Time Series. After validation, the Mean Percentage Error (MPE) has been calculated to prepare the Time Series. In the validation study OCEANSAT-2 OCM is showing good correlation when compared to MODIS CHL concentration Madhavan et al (2012) but OCEANSAT-2 OCM TS has not been available, since it was launched in the year 2009. Further the IRS P4 OCM (launched in the year 1999) CHL TS also is not a free product. Hence, MODIS CHL monthly data were considered for this research for the preparation of TS. The validation procedure and the MPE calculation were explained in detail in the following sections 4.3.3.1 to 4.3.3.7.

4.3.3.1 Introduction

Remote sensing of water quality parameters relies on the spectral properties of water leaving radiance Hinton (1991). The retrieval of CHL in oceanic waters, involves two major steps, firstly the atmospheric correction of visible channels to obtain normalized water leaving radiances and secondly the application of bio optical algorithm for retrieval of bio-physical parameters like chlorophyll- a concentration etc. An empirical algorithm (OC-2) has been proposed initially for SeaWiFS and OCM-1 ocean colour data by O’Reilly et al. (1998). This algorithm captures the inherent sigmoid
relationship between \( \text{Rrs490/Rrs555} \) band ratio (\( \text{Rrs}-\text{Remote sensing reflectance} \)) and CHL. This algorithm retrieves low as well as high chlorophyll concentration, signifying a better retrieval even in the case of ‘CASE-2 water’. Chauhan et al (2002a). The mathematical form of the algorithm with five coefficients is as shown below.

\[
C = 10^{(0.319 - 2.336 \times R + 0.879 \times R^2 - 0.135 \times R^3) - 0.071} \quad (4.1)
\]

For, \( 0.01 \text{ mg/m}^3 \leq C \leq 50 \text{ mg/m}^3 \),

Where \( C \) is the chlorophyll concentration in mg/m3

\[
R = \log_{10} \left[ \frac{\text{Rrs (490)}}{\text{Rrs (555)}} \right]
\]

In recent days the default algorithm for SeaWiFS is changed from OC2 to OC4 v4. The OCM-2 also adopts OC4 algorithm as default, but MODIS uses OC3 as default algorithm.

4.3.3.2 Study Area for Chlorophyll-a Validation

Figure 4.7 In-situ sampling site locations for CHL concentration in Bay of Bengal
The study area encompasses the CASE-1 and CASE-2 waters of Chennai, Kancheepuram, Vilupuram, Pondicherry, Cuddalore and Nagapattinam coastal districts in Tamil Nadu, covering a coastline length of 416 km. Tamil Nadu coast (Figure 4.7) falls in the latitudinal and longitudinal extensions of 8° 05’-13° 35’ N and 76° 15’-80° 20’ E on the Eastern part of Bay of Bengal. Tamil Nadu coast is the second longest coast line of 1076 km among the maritime states of India. The continental shelf area along the state is about 41,412 km².

4.3.3.3 Oceansat-2 OCM Data Product

OCM-2 Local Area Coverage L1B data for AUG 9th and 11th 2010 were obtained and processed using OCM enabled SEADAS software for Atmospheric correction to get Normalized water leaving radiance (nWL) Radiance. The OC4 algorithm (O’Reilly et al 1998, O’Reilly et al 2000) was used to convert the nWL into CHL in CASE-1 waters. This algorithm is similar to the two band OC3 algorithm developed for MODIS bands, and calculates CHL. However, the OCM-2 wavelength band configuration enables the OC4 algorithm to select the maximum band ratio from three ratios as opposed to the two ratios in the OC3 algorithm in a modified cubic polynomial equation O’Reilly et al (1998, 2000). Therefore, in the OC4 algorithm R is given as

\[ C = 10^{(0.366 - 3.067 \times R + 1.930 \times R^2 + 0.649 \times R^3 - 1)} \]  \hspace{1cm} (4.2)

Where C is the chlorophyll concentration in mg/l

\[ R = \log_{10} [\max \text{Rrs (443)}, \max \text{Rrs (490)}, \max \text{Rrs (510)} / \text{Rrs (555)}]. \]
4.3.3.4 MODIS Data Product

MODIS L1A data for July 17\textsuperscript{th}, Aug 09\textsuperscript{th} and Aug 12\textsuperscript{th} 2010 were obtained from the NASA Ocean Colour Discipline Processing System (OCDPS) at http://oceancolor.gsfc.nasa.gov/. The standard MODIS L1A ocean product is processed into L1B and finally to L2, to get CHL in CASE-1 and CASE-2 waters using OC3M algorithm (O’Reilly et al. 2000). MODIS images were initially processed using NASA software SeaDAS (version 6.1). The atmospheric effects were removed from the calibrated at-sensor radiance to obtain nWL (process from L1A to L1B) Then empirical OC3 inversion algorithm was applied for retrieval of CHL, which makes use of the following parameters like remote sensing reflectance, Rrs(\lambda), defined as the upwelling radiance leaving the water surface, Lu(\lambda), normalized by the downwelling irradiance Ed(\lambda), just above the water surface (process from L1B to L2). The OC3M algorithm (O’Reilly et al 2000) calculates CHL as

\[ C = 10^{(0.2830 + 2.753 \times R + 1.457 \times R^2 + 0.659 \times R^3 + 1.403R^4)} \]  \hspace{1cm} (4.3)

Where C is the chlorophyll concentration in mg/m3

\[ R = \log_{10} \left[ \max \text{Rrs (443)}, \frac{\max \text{Rrs (488)}}{\text{Rrs (550)}} \right] . \]

4.3.3.5 Cruise Data and Laboratory Chlorophyll - a Analysis

\textit{In-situ} CHL data were collected from two cruises (Figure 4.7) in BoB from 17 to 19 July 2010 and from 8 to 12 August 2010. Surface water samples of two litres each with a replica were collected between 9 A.M and 6 P.M with Nansen water sampler mounted on a rosette equipped with a SBE-SEABIRD - CTD. Samples were filtered with 47 mm Whatman GF/F glass fiber filters. As the water is being filtered, a few drops of suspension of magnesium carbonate have been added to prevent acidity on the filter. Finally CHL was determined by extracting pigments in 90 % acetone for 24
hours and measured by Strickland and Parsons (1972) method. During the cruises, SST, Secchi depth, salinity, Bottom depth and wind speed were measured along the ship track using SeaBird SBE19 CTD, Refractometer, Secchi Disc, Echo sounder and Automatic weather station. A total of 71 (30+41) station samples were taken up for the validation from both the cruises. The Nansen water sampler and the seawater filtration unit used for this analysis were shown in Figure 4.8.

Figure 4.8 (a) Nansen water sampler (b) Filtration unit

4.3.3.6 MODIS Chlorophyll-a Validation

The OCM-2 sensor has a temporal resolution of two days, while MODIS has a revisit time of 1 or 2 days. Because of different orbital geometries of OCM-2 and MODIS, sub satellite tracks of both sensors close to each other are taken into the comparison of two sensors CHL data. On AUG 09, 2010, the closest match between the sub satellite tracks of OCM-2 and MODIS could be obtained over the study area. The CHL values from all two cruises were grouped as exact date value, ± 24 hours value, CASE -1 water values and CASE -2 water values. All the measured in-situ data and the four grouped chlorophyll value data of OCM-2 and MODIS sensors are
correlated individually with *in-situ* data for validation and again between OCM-2 - OC2, OC3 & OC4 algorithms separately for the assessment and identification of better sensor, algorithm and timing for the CHL concentration in Bay of Bengal, India. For evaluation of chlorophyll, algorithms were evaluated through a regression analysis performed between satellite derived and *in-situ* measured CHL concentration values. Statistical parameters such as Coefficient of Determination ($R^2$) and RMSE are computed and used for evaluation.

\[
RMSE = \sqrt{\frac{\sum (\log(C_{i,mod}) - \log(C_{i,meas}))^2}{n}}
\]

(4.4)

Where $C_i$ is CHL concentration for a point/stations i, and n is the total no of stations in the data set. The performances of the algorithms were evaluated on the basis of a standard evaluation criterion suggested by Chauhan et al (2002b), Nagamani (2007) which states that Coefficient of Determination ($R^2$) should be more than 0.80 and RMSE error should be less than 0.185.

### 4.3.3.7 Mean Percentage Error (MPE)

The MPE is a relative measure of the forecasting error. It is subject to the “averaging” of the positive and negative errors. Positive and negative forecast errors can offset each other; as a result the formula can be used as a measure of the bias in the forecasts. A disadvantage of this measure is that it is undefined whenever a single actual value is zero.

\[
MPE = \left( \frac{\sum_{i=1}^{n} \left( \frac{\hat{Y}_i - Y_i}{Y_i} \right) \times 100}{n} \right)
\]

(4.5)
Where, \( Y_i \) is the observed value, \( \hat{Y}_i \) is the \textit{insitu} value to \( Y_i \), and \( n \) is the number of the observations of the validation set. \( Y \) is average mean value of the target. Negative value gives the under estimation of \textit{insitu} value and positive one gives the overestimation of \textit{insitu} CHL from this above formula. The MPE between the satellite and the observed value was -1.99588. This gives the under estimated results for satellite value for approximately two percent. So the entire MODIS extracted CHL value was added with additional two percent from their value for the preparation of MODIS Chlorophyll-a satellite Time Series. The validated TS also available in Annexure II. In addition to LTS and STS an additional TS component namely “Seasonality” is considered. Seasonal (S) component considers the ocean environmental parameters and fish catch of a particular month of every year for the entire period. Non-Seasonal (NS) refers to the all the values of parameters in the TS without specific reference to any month.

4.4 DEVELOPMENT OF NEURAL NETWORK STRUCTURE

4.4.1 Introduction

The traditional Auto-Regressive Integrated Moving Average (ARIMA \( p,d,q \)) models following Box- Jenkinson methodology with Multiple Linear Regression (MLR) normally used for obtaining wide variety of fisheries TS predictions. Monthly fish catch landings are generally nonlinear, which is not predicted effectively by -ARIMA models. However, to some extent it is efficient in modelling Linear phenomena in describing and predicting the fisheries TS of with a wide variety of species (Saila et al 1980), Hae-Hoon Park (1998), Stergiou KI (1996a), Stergiou KI (1996b), Stergiou et al (1997), Sathianandan (1995), Venugopal (1998)). Hence Artificial Neural Network (ANN) modelling has been chosen in this study for prediction analysis. The advantage of ANN over MLR models is the ability of ANN to directly take into account any non-linear relationships between the
dependent variables and each independent variable. Several authors have shown better performances of ANN as compared to the MLR (Ehrman et al 1996; Lek et al 1996b; Scardi 2001) in prediction capability.

Neural networks are a biologically inspired model, which tries to simulate the network of neurons in the human brain. The artificial neural networks consist of simple calculation elements, called neurons, and weighted connections between them called weights. There are many different types of neural networks depending on the architecture and training algorithms. The multilayer feed forward neural network is one of the most popular and commonly used network model Ripley (1994), with back-propagation learning algorithm, Rumelhart et al (1986). Typically, the feed-forward network contains three types of processing units (neurons), input units, output units and hidden units, organized in a hierarchy of layers: input layer, hidden layers and output layer.

ANN can be trained to perform complex functions by adjusting the values of the connections (weights) between the elements (neurons) according to one of several training algorithms. Back-propagation is the most popular training algorithm in which the training data propagated forward through the network and the output data are calculated. The error between the expected output and the calculated output is computed. Then a minimization procedure is used to adjust the weights between two connection layers starting backwards from the output layer to input layer. There are number of variations of minimization procedures that are based on different optimization methods, such as gradient descent, Quasi-Newton and Levenberg-Marquardt methods. The forward and backward propagation are executed iteratively over the training set until a stopping criterion is met. A typical three layer feed forward ANN is shown in Figure 4.9.
4.4.2 Feed Forward Back Propagation Neural Network

There are many variations of the back propagation algorithm. The simplest implementation of back propagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly i.e. the negative of the gradient. There are two different ways in which this gradient descent algorithm can be implemented as incremental mode and batch mode. In this study batch mode Back Propagation with \textit{traindm} training function was used for defining NNM, because the other default training function \textit{Levenberg-Marquard} is available in MATLAB was giving moderate convergence, speed and high MSE levels to the available data, when compared to \textit{traindm}. The formulas used in updating the synaptic weights of a multilayer feed forward neural network using the GD back propagation algorithm is explained in the following section. Srinivasan et al (2002).

4.4.3 Gradient Descent Back Propagation Algorithm

Let \((x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)\) represent the \(p\) vector - pairs used to train the network where \(x_i \in \mathbb{R}^N, y_i \in \mathbb{R}^M\). A back propagation feed forward network with an input layer, output layer and only one hidden layer is considered for analysis and simulation.
An input vector \( x_p = (x_{pl}, \ldots, x_{pn}) \) is applied to the input layer of the network. The input units distribute the values to the hidden layer units. 

The net input to \( j_{th} \) hidden unit is 

\[
net_{pi}^h = \sum_i w_{ji}^h x_{pi} + \theta_j^h
\]  

(4.6)

where 'h' superscript refers to the quantities on the hidden layer, \( w_{ji}^h \), is the weight on connection from \( i_{th} \) input unit to the \( j_{th} \) hidden unit, and \( \theta_j^h \) is the bias term. The output of this node is given by

\[
i_{pj} = f_j^h (net_{pi}^h)
\]  

(4.7)

where \( f \) is the activation function. The net input, and output for the \( k_{th} \) output node are

\[
net_{pk}^o = \sum_j w_{kj}^o i_{pj} + \theta_k^o
\]  

(4.8)

\[
o_{pk}^o = f_k^o (net_{pk}^o)
\]  

(4.9)

where the superscript 'o' refers to the quantities in the output layer, and \( w_{kj}^o \) is the weight on the connection between the \( j_{th} \) hidden unit and the \( k_{th} \) output unit.

The GD algorithm typically minimizes the function

\[
U(w) = d(w_{t+1}, w_t) + \eta L(y_t, o_t)
\]  

(4.10)

where \( d(w_{t+1}, w_t) = 1/2 |w_{t+1} - w_t|^2 \) is the squared Euclidean distance for all the components of the weight vectors, \( \eta \) is the learning rate, \( y_t \) is the desired
output at time $t$, and $o_t$ is the actual output of the algorithm at time $t$. Setting $\dot{U}(w) / \dot{w} = 0$ and using the squared Euclidean distance,

$$w_{t+1} - w_t + \eta L(y_t, o_t) = 0 \quad (4.11)$$

Which when rearranged results in

$$w_{t+1} = w_t + \eta L(y_t, o_t) \quad (4.12)$$

The direction in which to change the weights is determined by calculating the negative of the gradient of $L_{yt}$ with respect to the weights, $w_{t+1}$, $i$. Then the values of the weights can be adjusted such that the total loss is reduced. Thus, the gradient descent algorithm updates the weight vector by subtracting from it the gradient $\dot{L}_{yt}(o_t)$ multiplied by the scalar $\eta$.

The GD algorithm is applied to a back propagation algorithm based multilayer FFNN. The loss at a single neuron in the output layer is $(y_{pk} - o_{pk})$ where $y_{pk}$ is the desired output value and $o_{pk}$ is the actual output value from the $k_{th}$ unit, for the $p_{th}$ input. The loss minimized by the GD algorithm is the sum of the square of losses for all the output units where

$$\text{Total Loss: } L_P = 1 / 2 \sum_k (y_{pk} - o_{pk})^2 \quad (4.13)$$

The weight changes are proportional to the gradient of $L_p$ in the GD back propagation algorithm. The gradient of $L_p$ with respect to output layer weights is

$$\frac{\delta L_P}{\delta w_{kj}} = - (y_{pk} - o_{pk}) f_k^o (net^o_{pk}) i_p j. \quad (4.14)$$
If the output function \( f_k^o \) is linear, then

\[
\frac{\delta L_p}{\delta w_{kj}^o} = -(y_{pk} - o_{pk})i_{pj}. \tag{4.15}
\]

If the output function \( f_k^o \) is sigmoidal, then

\[
\frac{\delta L_p}{\delta w_{kj}^o} = -(y_{pk} - o_{pk})o_{pk}(1 - o_{pk})i_{pj}. \tag{4.16}
\]

Similarly the gradient of \( L_p \) with respect to hidden layer weights:

\[
\frac{\delta L_p}{\delta w_{ji}^h} = -\sum_k [(y_{pk} - o_{pk})f_k^o' (net_{pk}^o)w_{kj}^o f_k^h (net_{pj}^h)x_{pi}]. \tag{4.17}
\]

The weights on the output layer nodes are updated using

\[
w_{kj}^o (t + 1) = w_{kj}^o (t) - \eta \frac{\delta L_p}{\delta w_{kj}^o} \tag{4.18}
\]

i.e.,

\[
w_{kj}^o (t + 1) = w_{kj}^o (t) + \eta(y_{pk} - o_{pk})f_k^o (net_{pk}^o)i_{pj} \tag{4.19}
\]

The weights on the hidden layer nodes are updated using

\[
w_{ji}^h (t + 1) = w_{ji}^h (t) - \eta \frac{\delta L_p}{\delta w_{ji}^h} \tag{4.20}
\]

i.e.,

\[
w_{ji}^h (t + 1) = w_{ji}^h (t) + \eta\sum_k [(y_{pk} - o_{pk})f_k^o (net_{pk}^o)w_{kj}^o f_k^h (net_{pj}^h)x_{pi}]. \tag{4.21}
\]
4.4.3.1 Back Propagation with Training Function \textit{traingdm}

Traingdm is a network training function used in this study for all Multivariate NNM that updates weight and bias values according to gradient descent with momentum. Traingdm was a powerful batch algorithm for Feed Forward Back Propagation networks that often provides faster convergence when compared to other training functions. Traingdm, is a steepest descent with momentum, in which momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a low-pass filter, it allows the network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum. With momentum a network can slide through such a minimum. Traingdm can train any network as long as its weight, net input, and transfer functions have derivative functions.

Tiwari (2010) has said that ANN’s performance is more dependent on data representation than on the selection of learning algorithm. Hence, this study has focused on data pre processing to enhance the overall performance of neural networks trained by a gradient descent with momentum learning algorithm. In the error minimization of a gradient descent with momentum (GDM) learning algorithm, subsequent weighting factors are calculated in the steepest descent direction (negative of the gradient) as follows:

\[
W(k+1) = W(k) - \eta \nabla J(W) + \alpha \Delta W(k-1) \tag{4.22}
\]

where, \(w(k + 1)\) is the value of the weight vector at the iteration step \((k + 1)\), \(\eta\) is the learning rate adjusted at the \(k^{th}\) iteration, \(\alpha\) is the momentum coefficient and \(\nabla J(w)\) is the gradient of the performance index or the mean squared error specified as an averaged sum of instantaneous squared errors at the network output:
\[ \nabla J(W) = \frac{1}{zN} \sum_{n=1}^{N} \nabla E(W,n) \]  

(4.23)

The weight and bias updates are proportional to the performance index \( \nabla J \) by

\[ \nabla E(W,n) = \left[ \frac{\partial E}{\partial w_{i1}^h} \ldots \frac{\partial E}{\partial w_{i1}^h} \frac{\partial E}{\partial w_{i1}^o} \ldots \frac{\partial E}{\partial w_{im}^o} \right] \]  

(4.24)

where, \( N \) is the number of input and output vectors, \( n \) is the epoch number, \( z \) is the number of neurons at the output layer, \( \nabla E(w, n) \) is the gradient vector of the total instantaneous squared error at the network output that have components associated with the weights of the hidden and output layers \( W^h \) and \( W^o \), respectively:

\[ W^h = \left[ w_{i1}^h \ldots w_{ij}^h \ldots w_{ji}^h \right] j = 1, \ldots, s; i = 1, \ldots, k \]  

(4.25)

\[ W^o = \left[ w_{i1}^o \ldots w_{im}^o \ldots w_{im}^o \right] l = 1, \ldots, z; m = 1, \ldots, s. \]  

(4.26)

### 4.4.4 ANN Model Optimization

In this NNM, to model Sardine and Mackerel monthly prediction for the year 2010, the training parameters have been modified several times until the optimum performance has been achieved. In this part a number of modified training parameters will be presented along with justification for each case. Maximum number of iterations has been set to 5000 epochs, since the back-propagation with gradient descent momentum is showing some delayed convergence. For this optimization, initially the input layer consists of three input neurons (CHL, SST and PAR) for a three layered ANN with single output .i.e. 3-3-1.
Figure 4.10 Flow chart details of Neural Network methodology
A hidden layer having three neurons are considered for this optimization procedure and the last layer connects to the output variables and it is called the output layer having only one neuron (the target fish landing variable). After small number of iterations, the network converges to the optimum solution. Maximum validation failures have been set to 6 cases only since great number of failed validation cases may affect the network stability and generality, when new cases are presented to the network. Learning rate is used to enhance the training speed and efficiency. Momentum constant is an another factor to increase the training speed in ANN. Sections 4.4.4.7 and 4.4.4.8 gives the detailed explanation about Learning rate and Momentum constant optimization used in this study.

Figure 4.10 gives the methodology flow chat for the orientation of neural network optimisation, training and testing. A total of 112 Neural Network Model were developed by applying Remote Sensing inputs (CHL, SST and PAR) with its combinations are available in Table No.4.2. Normally NN having three phases during modelling like Training, Validation and Testing. A certain portion of TS has been allotted for each phase and their explanations were given in detail the coming section 4.4.4.1. In this study the Multivariate NNM on Sardine and Mackerel landings TS was modelled until all three training, validation, testing and its overall Regression should be above 0.5. Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship and zero means a random relationship. The number of neurons of the hidden layer is an important parameter of the network. In this type of network, the input layer is determined by the incoming signals. Hidden layer Neuron receives input signal from input layer neurons to which it is connected. Each of these connections has numerical weights associated with them.
Table 4.2 Details of different time series used for Neural Network Model Development

<table>
<thead>
<tr>
<th>SI.No</th>
<th>Input Parameters</th>
<th>Type of TS</th>
<th>CHL</th>
<th>SST</th>
<th>PAR</th>
<th>CHL-SST</th>
<th>CHL-PAR</th>
<th>SST-PAR</th>
<th>CHL-SST-PAR</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LONG TS MER Sardine NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LONG TS MER Sardine S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>LONG TS MER Mackerel NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>LONG TS MER Mackerel S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SHORT TS MER Sardine NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SHORT TS MER Sardine S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SHORT TS MER Mackerel NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SHORT TS MER Mackerel S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>SHORT TS MO Sardine NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SHORT TS MO Sardine S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>SHORT TS MO Mackerel NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SHORT TS MO Mackerel S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>SHORT TS MV Sardine NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SHORT TS MV Sardine S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SHORT TS MV Mackerel NS</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>SHORT TS MV Mackerel S</td>
<td>01 01 01 01 01 01 01 01 01 01 01 01 01 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL** | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 112 |

TS = Time Series; NS = Non Seasonal; S = Seasonal; MER = Merged CHL; MO = MODIS CHL; MV = MODIS Validated CHL.
These weights determine the nature and strength of the influence between the interconnected neurons. The signals from each input are then processed through a weighted sum on the inputs. The processed output signal is then transmitted to another neuron via a transfer function to produce an input signal to the output layer. The output layer processes its input signals in the same fashion.

Table 4.3 Details of Neural Network Model

<table>
<thead>
<tr>
<th>S. No</th>
<th>Descriptions</th>
<th>NN for single parameter</th>
<th>NN for two parameters</th>
<th>NN for all three parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>NN structure</td>
<td>3-4 layers</td>
<td>3 layers</td>
<td>3 layers</td>
</tr>
<tr>
<td>2.</td>
<td>Input to hidden Transformation</td>
<td>Tansig</td>
<td>Tansig</td>
<td>Tansig</td>
</tr>
<tr>
<td>3.</td>
<td>Hidden to output Transformation</td>
<td>Tansig</td>
<td>Tansig</td>
<td>Tansig</td>
</tr>
<tr>
<td>4.</td>
<td>Training function</td>
<td>Traingdm</td>
<td>Traingdm</td>
<td>Traingdm</td>
</tr>
<tr>
<td>5.</td>
<td>Trainparam.lr</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>6.</td>
<td>Trainparam.epochs</td>
<td>5000-500</td>
<td>5000-300</td>
<td>5000-225</td>
</tr>
<tr>
<td>7.</td>
<td>Trainparam.mc</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

4.4.4.1 Neural Network Training

During training the input and target were presented to the network and the network is adjusted according to its error. In this Neural Network a maximum of 156 (1998-2010) sample values were considered for single input Non-Seasonal and seasonal LTS architecture training. Out of which 144 values were considered for training purpose. Table 4.4 gives the details of training and testing sampling size to the entire modelling for both Seasonal
and Non-Seasonal. Training automatically stops when generalization stops improving, as indicated by an increase in the Mean Square Error of the validation samples. The number of neurons needs to be changed if the network does not perform well after training at one stage. This can be identified only by trial and error basis. Training multiple times will generate different results due to different initial conditions and sampling. Mean Squared Error is the average squared difference between outputs and targets (i.e. Lower values are better. Zero means no error).

The training phase is stopped when any of these conditions occurs:

- The maximum number of epochs (repetitions) is reached.
- The maximum amount of time is exceeded (not fixed in this modelling).
- Performance is minimized to the goal.(value zero set to this study)
- If the maximum fails exceeds six.

4.4.4.2 Neural Network Validation

These are used to measure network generalization, and to halt training when generalization stops improving. An amount of 20% of input sample values given for initial training was considered for validation purpose. All the training and validation sample values were taken randomly from the entire input value according to their percentage allocation.
Table 4.4  Details of Time Series sample sizes used in training and testing

<table>
<thead>
<tr>
<th>S. No</th>
<th>Name of Time Series (TS)</th>
<th>SST</th>
<th>CHL</th>
<th>PAR</th>
<th>Training and validation</th>
<th>No of samples</th>
<th>Testing year</th>
<th>No of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Time period of parameters</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.4.4.3 Neural Network Testing

In testing input and target values are not having any effect on training, when values are provided as an independent input variable for network performance during after training. In this NNM, the 2010\textsuperscript{th} year data i.e. January to December of 2012 sample values was given as target for testing to estimate the prediction of Sardine and Mackerel landings. After that the predicted values were again compared with the actual fish catch data to calculate their MSE value. The NNM was retrained until to arrive the minimum MSE value after repeated training and testing for each modelling. The details of training, validation and testing are available in Table No 4.4. For seasonal NNM also, the same procedure was adopted, where the training sampling number for LTS is eleven and for STS is four.

4.4.4.4 Transfer Function (TF) ‘tansig’

Transfer Function (TF) is a mathematical equation associated with the neurons. Typical TFs are polynomial (linear, quadratic, cubic, etc.), hyperbola (tanh, sigmoid), kernel (Gaussian) and wavelet. Depending upon the nature of the TF, different nets have been proposed. For example, Radial Basis Function (RBF) NN and probabilistic NN use radial and probability transfer functions, respectively. A neuron, in a given hidden layer, receives information from neurons of the previous layer. The TF operates on the information and produces the result, which is passed on to the next layer. During model development cconvergence has not achieved with maximum fails (6) allowed for training in NN with logsig TF. Further Hamid et al (2010) inferred that using tansig transfer function in output layer generated better output compared to logsig transfer function. Hence in this study also TanSig (TSG) transfer function has been applied. Figure 5.8 represents the tansig TF graphically.
Figure 4.11 Graphical representation of tansig Transfer Function

The formula used for the tansig TF is explained below

\[ tansig(n) = \frac{2}{1 + \exp(-2 \times n)} - 1 \]  

(4.27)

This is mathematically equivalent to \( \tanh(N) \). It differs in that it runs faster than the MATLAB implementation of \( \tanh \), but the results can have very small numerical differences. This function is a good trade off for neural networks, where speed is important and the exact shape of the transfer function is not. (MATLAB manual 2012).

4.4.4.5 Data Pre-processing and Normalization

Neural Network training can be made more efficient if certain pre-processing steps on the network inputs and targets are done. Several pre-processing techniques are usually applied before the data can be used for training to accelerate convergence. In multilayer networks, sigmoid transfer functions are generally used in the hidden layers. These functions become essentially saturated when the net input is greater than three \((\exp(-3) \equiv 0.05)\). If this happens at the beginning of the training process, the gradients will be very small, and the network training will be very slow. In the first layer of the network, the net input is a product of the input times the weight plus the bias. If the input is very large, then the weight must be very
small in order to prevent the transfer function from becoming saturated. It is standard practice to normalize the inputs before applying them to the network.

Generally, the normalization step is applied to both the input vectors and the target vectors in the data set. In this way, the network output always falls into a normalized range. The network output can then be reverse transformed back into the units of the original target data when the network is put to use in the field.

It is easiest to think of the neural network as having a pre-processing block that appears between the input and the first layer of the network and a post processing block that appears between the last layer of the network and the output, as shown in the following Figure 4.12.

![Figure 4.12 Graphical flow-chart representation of NN Pre-Processing](image)

In this study, mapminmax function is used to perform the pre-processing. If mapminmax is used to scale the targets, then the output of the network will be trained to produce outputs in the range \([-1, 1]\). To convert these outputs back into the same units that was used for the original targets, `ts`. Mapminmax settings do the calculation on each row of the matrix, as long as the inputs of NN are of the same type along the rows. Therefore, the xmax and xmin should be calculated separately for each row: The following code simulates the network that was trained, and then converts the network output back into the original units in MATLAB.
\[ an = \text{sim}(\text{net, pn}); \]
\[ a = \text{mapminmax}('reverse', an, ts); \]

The network output ‘\(an\)’ corresponds to the normalized targets ‘\(tn\)’. The unnormalized network output ‘\(a\)’ is in the same units as the original targets ‘\(t\)’. The algorithm used to apply for mapminmax function is given below

\[ y = (y_{\text{max}} - y_{\text{min}}) \frac{(x - x_{\text{min}})}{(x_{\text{max}} - x_{\text{min}})} + y_{\text{min}}; \quad (4.28) \]

It is assumed that \(X\) has only finite real values, and that the elements of each row are not all equal. (If \(x_{\text{max}} = x_{\text{min}}\) or if either \(x_{\text{max}}\) or \(x_{\text{min}}\) are non-finite, then \(y = x\) and no change occurs.). (MATLAB manual 2012)

### 4.4.4.6 BP Learning Rate (\(\mu\))

Selection of a value for the learning rate parameter, (\(\mu\)) has a significant effect on the network convergence. A high learning rate, \(\mu\) will accelerate training (because of the large step) by changing the weight vector, \(W\) significantly from one cycle to another. However, this may cause the search to oscillate on the error surface and never converge, thus increasing the risk of overshooting a near-optimal \(W\). In contrast, a small learning rate drives the search steadily in the direction of the global minimum, though slowly. A constant learning rate may be utilized throughout the training process. Wythoff (1993) suggests \(\mu = 0.1-1.0\) to his study. In this study, also a careful observation was made before assigning the learning rate to the model. The 3-layer ANN is initially tested with learning rate from 0.1 to 0.9. The learning rate of 0.3 gave a best minimum MSE which is allotted for all the models in this study. Table 4.4 gives the identification details for the selection of best Learning rate (0.3) for this study.
Table 4.5  Details of optimization of learning rate and momentum constant

<table>
<thead>
<tr>
<th>S.No</th>
<th>µ</th>
<th>Mc</th>
<th>No. of Epochs</th>
<th>5000</th>
<th>MSE</th>
<th>R</th>
<th>4000</th>
<th>MSE</th>
<th>R</th>
<th>3000</th>
<th>MSE</th>
<th>R</th>
<th>2000</th>
<th>MSE</th>
<th>R</th>
<th>1000</th>
<th>MSE</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MSE</td>
<td>R</td>
<td></td>
<td>MSE</td>
<td>R</td>
<td></td>
<td>MSE</td>
<td>R</td>
<td></td>
<td>MSE</td>
<td>R</td>
<td></td>
<td>MSE</td>
<td>R</td>
</tr>
<tr>
<td>1.</td>
<td>0.1</td>
<td>0.1</td>
<td></td>
<td>0.0084</td>
<td>0.6990</td>
<td>0.0075</td>
<td>0.7340</td>
<td>0.0082</td>
<td>0.6954</td>
<td>0.0064</td>
<td>0.6800</td>
<td>0.0080</td>
<td>0.0925</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0080</td>
<td>0.7367</td>
<td>0.0083</td>
<td>0.7156</td>
<td>0.0067</td>
<td>0.7124</td>
<td>0.0073</td>
<td>0.7543</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>0.3</td>
<td>0.6</td>
<td></td>
<td>0.0058</td>
<td>0.7540</td>
<td>0.0069</td>
<td>0.7570</td>
<td>0.0066</td>
<td>0.7019</td>
<td>0.0062</td>
<td>0.5900</td>
<td>0.0081</td>
<td>0.3037</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0067</td>
<td>0.7430</td>
<td>0.0055</td>
<td>0.7510</td>
<td>0.0086</td>
<td>0.7584</td>
<td>0.0085</td>
<td>0.7100</td>
<td>0.0061</td>
<td>0.6730</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>0.4</td>
<td>0.7</td>
<td></td>
<td>0.0092</td>
<td>0.7420</td>
<td>0.0091</td>
<td>0.7579</td>
<td>0.0068</td>
<td>0.7204</td>
<td>0.0084</td>
<td>0.7265</td>
<td>0.0080</td>
<td>0.1215</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0063</td>
<td>0.7460</td>
<td>0.0071</td>
<td>0.7443</td>
<td>0.0085</td>
<td>0.7014</td>
<td>0.0087</td>
<td>0.0352</td>
<td>0.0065</td>
<td>0.6212</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>0.2</td>
<td>0.5</td>
<td></td>
<td>0.0086</td>
<td>0.7352</td>
<td>0.0065</td>
<td>0.7222</td>
<td>0.0087</td>
<td>0.6867</td>
<td>0.0069</td>
<td>0.5644</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0073</td>
<td>0.6638</td>
<td>0.0080</td>
<td>0.7026</td>
<td>0.0067</td>
<td>0.6237</td>
<td>0.0065</td>
<td>0.6104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>0.4</td>
<td>0.5</td>
<td></td>
<td>0.0080</td>
<td>0.6139</td>
<td>0.0083</td>
<td>0.7897</td>
<td>0.0069</td>
<td>0.7345</td>
<td>0.0067</td>
<td>0.6980</td>
<td>0.0069</td>
<td>0.5632</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0078</td>
<td>0.7642</td>
<td>0.0064</td>
<td>0.7642</td>
<td>0.0078</td>
<td>0.7356</td>
<td>0.0081</td>
<td>0.7012</td>
<td>0.0081</td>
<td>0.5432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OF = OVER FIT;  Lr = LEARNING RATE;  µ=  MOMENTUM CONSTANT  
R = REGRESSION COEFFICIENT

4.4.4.7  BP Momentum Constant (µ)

Another way to increase the speed of convergence is to use a technique called momentum. A momentum term is commonly used in weight updating to help the search escape local minima and reduce the likelihood of search instability. µ accelerates the weight updates when there is a need to reduce µ to avoid oscillation. µ is set between 0 (no momentum) and values close to 1 (lots of momentum). A momentum constant of 1 results in a network that is completely insensitive to the local gradient and, therefore, does not learn properly. A µ > 1.0 yields excessive contributions of the
weight increments of the previous step and may cause instability. Conversely, an extremely small $\mu$ leads to slow training. Both a constant and adaptable momentum can be utilized. Wythoff (1993) suggests $\mu = 0.4 - 0.9$. Depending on the problem being solved, it seems that the success of training varies with the selected $\mu$ and a trial-and-error procedure is normally preferred. In this study, three layer ANN is initially tested with momentum constant ranging from 0.1 to 0.9. with different epochs. The $\mu$ value of 0.6 produced the best minimum MSE (0.0058) which is allotted for all the models in this study. Table 4.5 gives the identification details for the selection of best momentum coefficient (0.6) for this study. Balas et al (2010) using GDM learning algorithm, has obtained learning rate and momentum coefficients for his study as 0.3 and 0.6, respectively through trial and error. But the author has used uni-bipolar sigmoid as an TF for the hidden units and a linear function was employed for the output unit. In this study, both layers use tansig TF.

4.4.4.8 Number of Training Cycles

The number of training cycles required for proper generalization may be determined by trial and error. For a given ANN architecture, the error in both training and test data is monitored for each training cycle. Each iteration around the training cycle is known as an epoch. Either under training or overtraining can result in poor network performance. When training cycles (epochs) are insufficient, the network weights will not have adjusted sufficiently from their initial randomized states to maximize the performance of the network. However, it is also possible to over train the network, in which case the network will produce a very low error with the training set (indicating that the outputs given for this data set closely match the known results but will not perform well when making decisions based on other data; that is, the network loses its ability to generalize. Mark et al (1996). In this study, before finalizing the NNM, different epochs were given to the training
to achieve low MSE to finalise the training cycle. Table 4.4 gives the detailed description about the number of epochs applied for different training.

### 4.4.4.9 Hidden Layer Size

In most function approximation problems, one hidden layer is sufficient to approximate continuous functions Basheer et al (2000). Generally, two hidden layers may be necessary for learning functions with discontinuities. The determination of the appropriate number of hidden layers and number of Hidden Nodes (HN) in each layer is one of the most critical tasks in ANN design. Unlike the input and output layers, one starts with no prior knowledge as to the number and size of hidden layers. A network with too few hidden nodes would be incapable of differentiating between complex patterns leading to only a linear estimate of the actual trend. In contrast, if the network has too many hidden nodes it will follow the noise in the data due to over parameterization leading to poor generalization for untrained data. With increasing number of hidden nodes, training becomes excessively time-consuming. The optimal number of HN essential for network generalization may be a function of input / output vector sizes, the complexity of noise in the targets, size of training and test subsets, training algorithm, and, more importantly, the problem of non-linearity. Several rules of thumb are available in the literature which relates hidden layer size to the number of nodes in input and output. Facing exotic problems such as those with high nonlinearity, which normally forces us to try networks with hidden layers that may not match to any of these rules of thumb. The most popular approach to find the optimal number of hidden nodes is by trial and error with one of the above rules as starting point. Another way is to begin with a small number of hidden nodes and build on as needed to meet the model accuracy demand.

In general most Neural Network applications have only one or two hidden layers, since it is known that to approximate a set of reasonable
functions to a given accuracy, at most two hidden layers are needed Hertz et al (1991). The number of hidden layers and nodes in the hidden layers were determined by trial and error Czerwinski et al (2007). In this study also two hidden layers are used in LTS single input Neural Network analysis. Validation set was used repeatedly with the training set to determine all other remaining learning parameters like no of epochs, learning rate, momentum constant etc. The details of NNM are given in Table 4.5.

4.4.4.10 Convergence Determination

The best way to determine whether the network has reached the best set of weights for the training data is to validate the results using a validation set of data. In MATLAB randomly 20% of the input data are considered for validation. This is a separate data set that it do not use during training. Normally the network requires to generalize its inputs, so that it can correctly answer regression or classification queries not only for training data, but also for other examples. If the network trains too long, it will over fit to the training data, which means that it will correctly answer only examples that are in training data set. So, to help ensure that the network does not over fit, a cross validation procedure can be used. This involves training the network for a while (i.e., several epochs), and then presenting the network with the validation data. Again, the validation data is a set of data that the network has never seen before. Without updating weights, the error of network is showed, when presented with validation data. The process of training and validation is repeated, keeping track of the errors in both cases. It is easy to declare the convergence when the validation error is consistently decreasing after an initial increase. The weights of network are saved at this stage. In this way it is possible to identify the best performance in each model. Once convergence is confirmed, the saved weights (i.e., from when the validation error was the lowest) can be used as final network i.e. testing.
4.4.4.11 Different ANN Models

Subsequent to optimization the Neural Network analysis is initially started with one input and later with two and three input environmental parameters to develop NNM with different combinations of environmental parameters. The NN model is implemented by customising all optimisation and input parameters using scripts. In addition to different input combinations, the NN also incorporates with long and short TS orientation with their seasonality as Seasonal NN and Non-Seasonal NN. The details of data organization of different NNM were given in Table 4. A minimum of 10 repetitions is performed to identify the minimum MSE level for every iteration in testing for the best convergence. Czerwinski et al (2007) suggested a minimum of five repetitions to achieve best performance of NNM for testing with the same optimized training parameters. Sensitivity analysis is carried out for the model evaluation to find out the best model among them. The NNM identification process for three input three layer regression and its performance were given in Figures 4.13 and 4.14.

![Figure 4.13](image)

**Figure 4.13** Regression and training performance for 0.3 Lr & 0.6µ at 5000 epochs
4.5 METHODS OF EVALUATION

Several measures of accuracy were calculated in the calibration between model output and observed value. A measure of correlation between the observations and the predictions is the coefficient of correlation. The proportion of the total variance in the observed data that can be explained by the model was described by the coefficient of determination ($R^2$). The estimators to quantify the errors in the same units of the variance were the square root of the mean square error (RMSE), and Mean Absolute Error (MAE). On the other hand, other measures of variance such as Coefficient of Efficiency (E) Nash and Sutcliffe (1970); Kitanidis and Bras (1980), the Average Relative Variance (ARV) Grino (1992), and the Percent Standard Error of Prediction (SEP) Ventura et al (1995) are also analyzed for
sensitivity analysis in this study. The E and AVR were used to see how the models explain the total variance of the data and represent the proportion of variation of the observed data considered for NNM. The SEP allows the comparison of the prediction from different models and different problems because its dimensionless. For a perfect performance, the values of $R^2$ and E should be close to one and those of SEP and ARV close to zero. The optimal model is selected when RMSE and MAE are minimized. The above estimators are explained in the following sections 4.5.1 to 4.5.8

4.5.1 Coefficient of Correlation ($r$)

The relationship between two variables such that a change in one variable results in a positive or negative change in the other variable and also a greater change in one variable result in corresponding greater or smaller change in the other variable is known as correlation. It measures the strength and the direction of a linear relationship between two variables. If $x$ and $y$ have a strong positive linear correlation, $r$ is close to +1. An $r$ value of exactly +1 indicates a perfect positive fit. Positive values indicate a relationship between $x$ and $y$ variables such that, as values for $x$ increase, values for $y$ also increase. If $x$ and $y$ have a strong negative linear correlation, $r$ is close to -1. An $r$ value of exactly -1 indicates a perfect negative fit. Negative values indicate a relationship between $x$ and $y$ such that as values for $x$ increase, values for $y$ decrease. If there is no linear correlation or a weak linear correlation, $r$ is close to 0. A value near zero means that there is a random, nonlinear relationship between the two variables.

The mathematical formula for computing ‘$r$’ is

$$r = \frac{n \sum Y_i \hat{Y}_i - \left( \sum Y_i \right) \left( \sum \hat{Y}_i \right)}{\sqrt{n \left( \sum Y_i^2 \right) - \left( \sum Y_i \right)^2} \sqrt{n \left( \sum \hat{Y}_i^2 \right) - \left( \sum \hat{Y}_i \right)^2}}$$

(4.29)
In the above formula, $y_i$ is the observed value, $\hat{y}_i$ is the predicted value to $y_i$, and $n$ is the number of the observations of the validation set. $\bar{y}$ is average mean value of the target.

4.5.2 Coefficient of Determination ($R^2$)

The Coefficient of Determination is the square of the Coefficient of Correlation. It is a useful and readily comprehensible measure for indicating the percentage variation in the dependent variable which is accounted for by the independent variable. In other words the Coefficient of Determination gives the ratio of the explained variance to the total variance. The coefficient of determination, represents the percent of the data that is the closest to the line of best fit. For example, if $r = 0.922$, then $r^2 = 0.850$, which means that 85% of the total variation in $y$ can be explained by the linear relationship between $x$ and $y$. The other 15% of the total variation in $y$ remains unexplained. The coefficient of determination, is a measure of how well the regression line represents the data. If the regression line passes exactly through every point on the scatter plot, it would be able to explain all of the variation. The further the line is away from the points, the lesser it is able to explain.

4.5.3 Coefficient of Efficiency ($E$)

Nash–Sutcliffe efficiencies can range from $-\infty$ to 1. An efficiency of 1 ($E = 1$) corresponds to a perfect match of modelled landing data to the observed data. An efficiency of 0 ($E = 0$) indicates that the model predictions are as accurate as the mean of the observed data, whereas an efficiency less than zero ($E < 0$) occurs when the observed mean is a better predictor than the model or, in other words, when the residual variance (described by the numerator in the expression above), is larger than the data variance (described
by the denominator). Essentially, the closer the model efficiency is to 1, the more accurate the model is.

The mathematical formula for computing $E$ is

$$E = 1.0 - \frac{\sum_{i}^{n} |Y_i - \hat{Y}_i|^2}{\sum_{i}^{n} |Y_i - \bar{Y}|^2}$$

(4.30)

### 4.5.4 Average Relative Variance (ARV)

The Average Relative Variance is the remaining portion of the Coefficient of Efficiency, which should be always close to zero. The formula representing the Average Relative Variance is as follows.

$$\text{ARV} = 1.0 - E$$

(4.31)

### 4.5.5 Root Mean Square Error (RMSE)

The RMSE analysis provides useful information of the accuracy between predicted value of a model and observed value. The RMSE statistics for the comparison of the two data generated using the following formula. These individual differences are called residuals, when the calculations are performed over the data sample that was used for estimation, and are called prediction errors when computed out-of-sample. The RMSE serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{n}}$$

(4.32)
4.5.6 Mean Absolute Error (MAE)

Mean Absolute Error is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. The MAE is a linear score which means that all the individual differences are weighted equally in the average. The formula used to calculate MAE is

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_i|
\]

(4.33)

4.5.7 Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) is the mean or average of the absolute percentage errors of forecasts. Error is defined as actual or observed value minus the forecasted value. Percentage errors are summed without regard to sign to compute MAPE. The absolute values of all the percentage errors are summed up and the average is computed. This measure is easy to understand because it provides the error in terms of percentages

\[
MAPE = \frac{100}{n} \times \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|
\]

(4.34)

When having a perfect fit, MAPE is zero. But in regard to its upper level the MAPE has no restriction. In general a MAPE of 10% is considered very good, a MAPE in the range 20% - 30% or even higher is quite common.
4.5.8 Standard Error of Prediction (SEP)

Standard error is a statistical term that measures the accuracy with which, a sample represents a population. In statistics, a sample mean deviates from the actual mean of a population and this deviation is the standard error.

\[
% \text{ SEP} = 100 \frac{RMSE}{Y} \quad (4.35)
\]

A value quantifying the uncertainty of a prediction; mathematically, the square-root of the prediction variance. (The prediction variance is the variation associated with the difference between the true and predicted value.) As a rule, 95 percent of the time, the true value will lie within the predicted value plus or minus two times the prediction standard error if data is normally distributed. So SEP should be as low as possible.

4.6 DECISION SUPPORT SYSTEM FOR SARDINE AND MACKEREL FISH ABUNDANCE

A common definition of DSS is: “A Decision Support Systems (DSS) is a class of information systems (including but not limited to computerized systems) that support business and organizational decision-making activities. A properly designed DSS, is an interactive software-based system intended to help decision makers compile useful information from a combination of raw data, documents, personal knowledge, or business models to identify and solve problems and make decisions”.

The main objective of the spatial analysis tools used in this study, is to identify the potential fish harvesting zones within the study area for the Sardine and Mackerel species. The spatial variation of fish abundance is
calculated based on a combination of parameters like SST, CHL, and Depth of the Water and the Seasonality of the Year on pixel basis.

CHL, SST and Depth data sets of 4.5x4.5 Km resolution were co-registered for the purpose of determining index based fish abundance by assigning ranks and weights. The ranks for each layer is assigned based on prior knowledge about species habitat Antony Raja et al (1969). Depth, SST and CHL preference. Prathibha Rohit et al (2004); saitoh et al (2008); Semedi et al (2009). In order to reduce subjectivity in assignment of ranking and weights Analytical Hierarchy Process (AHP) is used in this study. Pair wise comparison matrix was generated for both Sardine and Mackerel separately. The weights were derived from NNM and based on these ranks and weights pixel based raster overlay analysis was carried out.

4.6.1 Study Area for DSS

The study area is Nagapattinam coastal district situated in Tamil Nadu, along east coast of India in Bay of Bengal with a coast line length of 190 km. The Navigational chart (No.7706) issued by National Hydrographic Office, Government of India, in the scale of 1:3,500,000 is used for preparing the base map and isobaths preparation. The western boundary of study area is defined by the 1000m isobath in the ocean. The study area in detail is shown in Figure 4.15. The study area comprises of five coastal taluks namely Nagapattinam, Sirkazhi, Tarangampadi, Mayiladuturai and Vedaranniyam. A total of 44 major and minor landing canters are available in the study area for the collection of Sardine and Mackerel landings.
Figure 4.15 Location map of the study area with 1000m isobaths for DSS

4.6.2 Satellite Data

The main purpose of this DSS development is to develop the Species Specific Potential Fishing Zone (SSPFZ) from the freely available ocean colour images. The CHL and SST image used in this study is part of freely available global dataset (http://oceancolor.gsfc.nasa.gov/cgi/l3). In Hierarchical Data Format (HDF) and Standard Mapped Index (SMI) format, with a spatial resolution of 4.5x 4.5 Km. The global dataset also provides three day images of CHL and SST, which could be used when the daily data is not available.
4.6.3 Bathymetry Data

Bathymetry images were generated from General Bathymetric Chart of the Oceans (GEBCO) data set for the study area. The original resolution of GEBCO was 900x900m. Since the CHL and SST images are having a spatial resolution of 4.5 Km, the GEBCO image also was converted into 4.5x 4.5Km resolution in ARC-GIS environment and co-registered.

4.6.4 Methodology

This study demonstrate the use of freely available global dataset of ocean parameters processed in open source GIS environment, making it economically viable to be implemented for vast coastal stretches. For the purpose of delineating SSPFZ, spatial variability of fish abundance is expressed as a function of CHL, SST and Depth (bathymetry). Based on NNM it is concluded that PAR is not a significant parameter in predicting fish abundance. However, several studies have shown that George et al (2012); Prathibha Rohit et al (2004) depth as a significant parameter and since it does not vary temporally (except near shore). It is derived from General Bathymetric Chart of the Oceans (GEBCO) data set.

4.6.5 Determination of Ranks and Weights

Based on the information from the Neural Network analysis the weight of CHL and SST were derived. The weightages are derived from the best performed TS model prediction level evaluated with $R^2$ in the study area. The weightage earned by the parameters again classified with their relative importance, identified through references and expert opinions for the development of ranking in AHP.
In order to have a rational assignment of ranks for various of CHL, SST and Depth parameters and their weights, it is proposed to use Multi Criteria Evaluation (MCE). In general, MCE seeks to identify the alternatives or options that are to be investigated and decided upon, a set of criteria by which to rank these alternatives and the method by which the alternatives are to be ranked and preferences aggregated. MCE is a process for combining spatial data according to their importance in making a given decision.

In this study, Analytic Hierarchy Process (AHP), a powerful MCE process was selected for derivation of ratings and weights of parameters, because of its simple hierarchical structure, sound mathematical basis, widespread usage and its ability to measure inconsistency in judgments.

4.6.5.1 Analytic Hierarchy Process

AHP is a MCE process developed by Saaty (1980) for decision making when confronted with conflicting and qualitative criteria. These are two main advantages of AHP. Firstly, it uses a scoring model, which allows an evaluator to subjectively compare qualitative and quantitative aspects together. Secondly, it contains a mechanism by which inconsistency in subjective comparison is avoided and estimated. The following are the major steps involved in solving a decision-making problem using AHP.

**Step 1:** Decompose the problem into hierarchy of elements, namely criteria an sub-criteria (i.e. criteria are the three parameters in the DSS and sub-criteria are the ranges within each parameter). The criteria and sub-criteria hierarchy ensures that the alternatives are appropriately related to the overall decision-making process.
**Step 2:** At each hierarchical level, each criterion is compared with one another referred to as pair wise comparison of criteria, resulting in a Pair wise Comparison Matrix (PCM). For example, to generate PCM for Ranges of CHL at level 2, the five ranges of CHL described in Table 5.7 are compared with one another. This results in a PCM for ranges of CHL with the order of 5 x 5. The elements of PCM are expressed as relative importance of one range (i.e. ratio of ratings of these ranges) over the other range. If criteria \( i \) is \( n \) times as important as \( j \), then \( j \) is \( 1/n \) times as important as \( i \). Hence, the PCM is reciprocal in nature, i.e. the upper diagonal elements are reciprocal of lower diagonal elements. And also all the diagonal elements will be 1 as the same criterion is compared against it; The relative importance is to be assigned a value in a scale of 1-9 as suggested by Saaty (1980). The Saaty’s scale of pair wise comparisons is given in Table 4.6. At level 1, the parameters themselves are compared against one another in terms of relative importance of their weights.

**Table 4.6 Saaty’s scale of pair wise comparison of criteria**

<table>
<thead>
<tr>
<th>SI. No.</th>
<th>Intensity of Importance</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Equal Importance</td>
<td>Two criteria contribute equally to the objective</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>Week Importance of one over other</td>
<td>Experience and judgement slightly favour one criteria over the other</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>Essential or strong importance</td>
<td>Experience and judgement strongly favour one criteria over the other</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>Very strong or demonstrated importance</td>
<td>A criteria is favoured very strongly over another</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>Absolute importance</td>
<td>The evidence favouring one criteria over another is of the highest possible order of affirmation</td>
</tr>
<tr>
<td>6</td>
<td>2,4,6,8</td>
<td>Intermediate values between adjacent scale values</td>
<td>When compromise is needed</td>
</tr>
</tbody>
</table>
Step 3: The relative importance of each criterion in the PCM is used to arrive at the ratings in level 2 and weights in level 1. If there are $n$ criteria, then PCM of order $n \times n$ can be written as follows:

\[
\begin{pmatrix}
\frac{W_1}{W_1} & \frac{W_1}{W_2} & \ldots & \frac{W_1}{W_n} \\
\frac{W_2}{W_1} & \frac{W_2}{W_2} & \ldots & \frac{W_2}{W_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{W_n}{W_1} & \frac{W_n}{W_2} & \ldots & \frac{W_n}{W_n}
\end{pmatrix}
\begin{pmatrix}
W_1 \\
W_2 \\
\vdots \\
W_n
\end{pmatrix}
= \begin{pmatrix}
nW_1 \\
nW_2 \\
\vdots \\
nW_n
\end{pmatrix}
\]

In matrix form, it is written as follows:

\[
A^{(i,j)} W^{(i,j)} = nW^{(i,j)}
\] (4.36)

Where $A$ is $n \times n$ PCM in terms of ratio of ratings/weights:

$W$ is the ratings/weights (priority rankings/weights) of criteria: and

$n$ is the order of the PCM matrix

The input matrix is ‘$A$’ and the solution for Equation 4.1 is the common eigen value problem. The solution to this set of equations is found by solving an $n$ order equation and the details are given below:

- Raise the PCM to powers that are successively squared each time
- The rows of the PCM and the squared PCM are normalized (i.e. each row element is divided by the sum of the elements in that row) to give the eigen vectors of PCM and the raised PCM
- The difference between the eigen vectors is checked against a threshold value (i.e. 0.0001) and if the difference is less than the threshold value, the eigen vectors of the raised PCM is the estimated rating/weight vector.

**Step 4:** The PCM is strongly consistent if the following condition is true:

\[ A_{ik} \times A_{kj} = A_{ij} \text{ for all } i,j \quad (4.37) \]

It can be shown that for any matrix, small perturbations in the entries (PCM) elements) imply similar perturbations in the eigen values. Thus, the eigen value problem for the inconsistent case can be written as

\[ [A] \times [W] = \lambda_{\text{max}} [W] \quad (4.38) \]

Where \( \lambda_{\text{max}} \) will be close to \( n \) (actually greater than or equal to \( n \)) and the others will be close to zero. The closer the \( \lambda_{\text{max}} \) to \( n \), the more consistent is the judgements in PCM elements. Hence, the difference, \( \lambda_{\text{max}} - n \), can be used as a measure of inconsistency. However, Saaty (1980) defined Consistency Index (CI), as a measure to indicate consistency of judgements, as follows:

\[ CI = \frac{\lambda_{\text{max}} - n}{(n-1)} \quad (4.39) \]

In the above equation, CI is divided by (\( n-1 \)) as it represents the average of the remaining eigen values. The first component of ‘nw vector’ is divided by the first component of the estimated ‘w vector’ and the same is repeated for all the elements in the ‘nw vector’. These values are summed up and divided by the number of criteria (i.e. order of the matrix), to estimate the value of \( \lambda_{\text{max}} \). In equation form the \( \lambda_{\text{max}} \) can be written as:
\[ \hat{\lambda}_{\text{max}} = \frac{\left( \frac{nW_1}{W_1} \right) + \left( \frac{nW_2}{W_2} \right) + \cdots + \left( \frac{nW_n}{W_n} \right)}{n} \]  
\[ (4.40) \]

**Step 5:** In order to derive a meaningful interpretation of CI, Saaty (1980) simulated random pair wise comparisons for different size matrices (i.e. up to the order of 15), calculating the CI, and arriving at an average CI for random judgements for each size of the matrix. This average CI is referred to as Random Index (RI) and the values are given in Table 4.7 (Saaty 1980). The ratio of the CI to RI is defined as Consistency Ratio (CR).

Table 4.7 Consistency ratio values of matrices of various orders

<table>
<thead>
<tr>
<th>Matrix Size</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>RI</td>
<td>0.00</td>
<td>0.00</td>
<td>0.58</td>
<td>0.90</td>
<td>1.12</td>
<td>1.24</td>
<td>1.32</td>
<td>1.41</td>
<td>1.45</td>
<td>1.49</td>
<td>1.51</td>
<td>1.48</td>
<td>1.56</td>
<td>1.57</td>
<td>1.59</td>
</tr>
</tbody>
</table>

Since a set of perfect consistent judgments produces a CI of 0, the CR will also be 0. A CR value less than 0.1 is considered to be acceptable and indicates that the judgements are made consistently.

**Step 6:** If the CR value is greater than 0.1, it indicates serious inconsistency the judgements made while creating PCM. In such cases, the judgements made in determining the PCM elements are to be modified so as to remove inconsistency. The element of PCM whose value is to be modified is determined by the following condition:
PCM element to be corrected = Maximum of (|A_{ij} - W_i/W_j|) for all I, j \hspace{1cm} (4.41)

**Step 7:** The process between steps 1 and 5 is repeated till the CR value is less than 0.1.

**Step 8:** The process is repeated for all criteria and sub-criteria at levels 1 and 2 to determine the ratings of the ranges of parameters and weights of parameters respectively.

### 4.6.6 DSS Model Development

The software QGIS has been selected to implement DSS as it is an open source GIS software available with free license. Subsequent to the determination of ranks to the parameters from AHP, the next step is to create the index based Fish Abundance Index (FAI). The index based FAI tool is implemented by customising QGIS using scripts. The FAI tool requires CHL, SST and Depth images at 4.5km resolution which are co-registered. The main advantages of the FAI tool are as follows:

- It uses freely available global dataset of CHL, SST and Depth images
- It is implemented in open source GIS environment for the development of FAI model

The conceptual FAI model is shown in detail in Figure 4.16.
CHL, SST and depth images are clipped to AOI (Area of Interest, i.e. study area) and co-registered to facilitate raster based overlay analysis. At 4.5x4.5Km grid resolution, the study area is covered by 406 pixels.

The HDF format of global dataset requires preprocessing tasks and the very beginning of the script, a pre processing check has been assigned to check for overlapping of pixels between the layers. This ensures a match up of the three layers so that the corresponding values of SST, Chlorophyll and Depth of a pixel can be obtained. The detailed explanation of the input data arrangement and the pre-processing check is shown in flowchart in the Figure. 4.17.

Figure 4.16 Flowchart showing conceptual diagram of FAI model

\[
FAI = (P_{sst} + P_{chl} + P_{dep})
\]
The next step in the FAI model was data processing and analysis. Raster pixels supports spatial analysis using a high-level computation language. Thus, processing between layers utilizes a simple and efficient map-algebra calculation of numeric cell values, which are executed through scripts developed for that purpose in QGIS.

The next step is assigning ranks to the pixels with the information available in it. Then the weight and ranks of each pixel are multiplied to get the corresponding pixel value. Finally all the three layer information was summed to arrive at the net score of all pixels as given in Equation 4.42.
(sum of pixel values of SST, Depth and Chlorophyll). The net score is the Fish Abundance Index (FAI), which is derived individually for both Sardine and Mackerel species. The SSPFZ is arrived at, in terms of FAI, which indicates the spatial variability of fish abundance. The ranks of Sardine and Mackerel species are determined separately by using different pairwise comparison matrices. The weights are derived from NNM analysis as given in Figure 4.18 shows the procedure followed for the development of FAI. The FAI is calculated as

\[
FAI = (P_{sst} + P_{chl} + P_{dep})
\]  

(4.42)

**Figure 4.19 Flow chart details of DSS weight assignment and output**

The FAI (net score) is categorized into three categories namely, High (FAI-1), Moderate (FAI-2), and low (FAI-3) using natural breaks classification method. The combined FAI-1 and FAI-2 zones which has higher values of FAI, is proposed as the fishing ground that fishermen should target for improved fish catch.
4.6.7 Validation of SSPFZ

In this study, two types of validation procedures are evaluated. First with the weekly data collected from the landing centres and the next one is with the current INCOIS PFZ advisory map.

For the first method of validation weekly Sardine and Mackerel landing data pertaining to month of July is aggregated compared with the FAI-1 derived for the respective week. i.e FAI-1, which is having higher values in the natural break classification method is summed up for all pixels in that week and compared with the Sardine and Mackerel weekly catch.

For the second validation, the current PFZ advisory with the PFZ line available in the study area was retrieved from INCOIS, with a minimum of 2 maps per month. Since the current PFZ advisory indicate PFZ as a linear feature for the purpose of comparison, using proximity analysis, three types of buffers (4.5 x 4.5, 9.0x9.0 and 13.5 x13.5Km) on either side of PFZ were created to see the categories of FAI (High, Moderate and Low) present in the buffers. The buffer distance of 4.5 km is arrived to match with the spatial resolution of satellite images.

The higher number of FAI-1 and FAI-2 categories pixels, which are present in the 4.5 km buffer zone, indicate that higher FAI values are in close proximity with the PFZ. Similarly, higher number of FAI-3 and FAI-2 pixels in 13.5 km buffer indicates that they are farther away from PFZ. The statistical details in terms of number of pixels, their sum and mean are tabulated and analysed for validation purposes. A total of 22 PFZ advisory were collected (i.e two per month) and the SSPFZ is validated with these PFZ, in terms of spatial categories of FAI.