DAMAGE LEVEL PREDICTION OF NON-RESHAPED BERM BREAKWATER USING SOFT COMPUTING TECHNIQUES

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

By

HARISH N.

DEPARTMENT OF APPLIED MECHANICS AND HYDRAULICS

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,

SURATHKAL, MANGALORE – 575 025

JULY 2014

D E C L A R A T I O N

I hereby *declare* that the Research Thesis entitled "**Damage Level Prediction of Non-Reshaped Berm Breakwater using Soft Computing Techniques**" which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Civil Engineering** is a *bonafide report of the research work carried out by me*. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

100533AM10P01, Harish N.

(Register Number, Name and Signature of the Research Scholar)

Department of Applied Mechanics and Hydraulics

Place: NITK- Surathkal

Date: 08. 07. 2014

C E R T I F I C A T E

This is to *certify* that the Research Thesis entitled "**Damage Level Prediction of Non-Reshaped Berm Breakwater using Soft Computing Techniques**" submitted by **Harish N.** (Register Number: AM10P01) as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

Prof. Subba Rao
(Research Guide)

Dr. Sukomal Mandal
(Research Guide)

(Chairman – DRPC)

DEPARTMENT OF APPLIED MECHANICS AND HYDRAULICS
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE –575 025

ACKNOWLEDGEMENTS

A successful project is a dream comes true for any student and to fulfill this dream, efficient teamwork is required. I am thankful to a lot of people who are directly and indirectly involved in this project.

The satisfaction that accompanies the successful completion of the project work would be incomplete without the mention of those people who made it possible through their guidance, support and encouragement.

It gives me a great pleasure to acknowledge and sincerely express my deep sense of gratitude to my research supervisors Prof. Subba Rao, Professor Department Applied Mechanics and Hydraulics, NITK and Dr. Sukomal Mandal, Chief Scientist, Head of Ocean Engineering Division, National Institute of Oceanography for their encouraging, logical and constructive suggestions in the execution of this project work. The interaction and the time spent in discussions are imprinted in my memory permanently. Only with their moral support and stimulating guidance, this research work could be completed and I could publish my work in many international journals and conferences.

I express my deep sense of gratitude to the Director of National Institute of Oceanography for permitting me to carry out my research work and to make use of institutional infrastructure facilities.

I am greatly indebted to Research Progress Appraisal Committee members, Prof. M. C. Narasimhan and Prof. S. M. Kulkarni, for their useful suggestions during the progress of the work.

I sincerely acknowledge the help and support of Prof. M. K. Nagaraj, former Head of the Department and Prof. Subba Rao, present Head of the Department of Applied Mechanics and Hydraulics, NITK, Surathkal, for permitting me to use the departmental computing and laboratory facilities and his continuous support in completing the work.

I express my deep sense of gratitude to the Research Scholar Mr. Prasanth J. who helped me in every mode of action to complete my research work successfully. Mr. Lokesha, Mr. Rakesh R., Dr. Aneelraj, Kumar raju and Karthika who inspired me to carry out my research work.

I sincerely acknowledge the help and support of all the Professors, Associate Professors, Assistant Professors and Teaching and non teaching staff of Department of Applied Mechanics and Hydraulics, NITK, Surathkal, in completing the work.

I gratefully acknowledge the support and all help rendered by Dr Balakrishna Rao for the experimental data. I gratefully acknowledge the support of all the friends during the research work.

I sincerely acknowledge the invaluable help rendered by Dr. Sanjay Govind Patil, Incharge Head, Caledonian College of Engineering, Muscat, Oman.

I sincerely acknowledge Mr. B. Jagadish, Foreman, Department of Applied Mechanics and Hydraulics, and Mr. Balakrishna, Literary Assistant at NITK, Surathkal, for their support and help during the research work.

I express my heart felt gratitude to the authors of all those research publications which had been referred to in preparing this thesis.

Finally, I wish to express gratitude, love and affection to my beloved family members, father Sri. Narayana, mother Smt. Lakshmi, brothers Sri Satheesh N. & Varada N., sister Smt. Asha, bother-in-law Sri Dinesh B.K. and wife Smt. Vindhya for their encouragement, moral support and all their big and small sacrifices on the road to the completion of my research.

Harish N.

ABSTRACT

Tranquility condition inside the port and harbor has to be maintained for loading cargo and passengers. In order to maintain calm condition inside the port and harbor, breakwater has to be constructed to dissipate wave energy that is coming inside. The alignment of the breakwater must be carefully considered after examining the predominant direction of approach of waves and winds, degree of protection required, magnitude and direction of littoral drift and the possible effect of these breakwaters on the shoreline. In general these studies are invariably conducted in a physical model test where various alternatives are studied and the final selection will be based on performance consistent with cost. Considering the coastal boundary and depth variation, field analysis of wave structure interaction, determination of stability and damage level of berm breakwater structure is difficult. Mathematical modeling of these complex interactions is difficult while physical modeling will be costly and time consuming. Hence one has to depend on physical model studies which are expensive and time consuming.

Soft computing techniques, such as, Artificial Neural Network (ANN), Support Vector Machine (SVM),Adaptive Neuro-Fuzzy Inference System (ANFIS) and Particle Swarm Optimization (PSO) have been efficiently proposed as a powerful tool for modeling and predictions in coastal/ocean engineering problems. For developing soft computing models in prediction of damage level of non-reshaped berm breakwater, data set are obtained from experimental damage level of non-reshaped berm breakwater using regular wave flume at Marine Structure Laboratory, National Institute of Technology, Karnataka, Surathkal, Mangalore, India. These data sets are divided into two groups, one for training and the other for testing. The input parameters that influence the damage level (S) of non-reshaped berm breakwater, such as, relative wave steepness (H/L_0), surf similarity (ζ), slope angle ($\cot\alpha$) relative berm position by water depth (h_B/d), relative armour stone weight (W_{50}/W_{50max}), relative berm width (B/L_0) and relative berm location (h_B/L_0) are considered in developing soft computing models for prediction damage level.

The ANN model is developed for the prediction of damage level of non-reshaped berm breakwater. Two network models, ANN1 and ANN2 are constructed based on the parameters which influence the damage level of non-reshaped berm breakwater. The seven input parameters that are initially considered for ANN1 model are (H/L_0), (ζ), (cot

α), (h_B/d), (W_{50}/W_{50max}), (B/ L_0) and (h_B/L_0). The ANN1 model is studied with different algorithm namely, Scaled Conjugate Gradient (SCG), Gradient Descent with Adaptive learning (GDA) and Levenberg-Marquardt Algorithm (LMA) with five numbers of hidden layer nodes and a constant 300 epochs. LMA showed good performance than the other algorithms. Also, influence of input parameters is evaluated using Principal Component Analysis (PCA). From PCA study, it is observed that cotα is the least influencing parameter on damage level. Based on the PCA study, least influencing parameter is discarded and ANN2 model is developed with remaining six input parameters. Training and testing of the ANN2 network models are carried out with LMA for different hidden layer nodes and epochs. The ANN2 with LMA 6-5-1 with 300 epochs gave good results. It is observed that the correlation of about 88% between predicted and observed damage level values by the ANN2 network models and measured values are in good agreement

Furthermore, to improve the result of prediction of damage level of non-reshaped berm breakwater, SVM model was developed. This technique works on structural risk minimization principle that has greater generalization ability and is superior to the empirical risk minimization principle as adopted in conventional neural network models. This model was developed based on statistical learning theory. The basic idea of SVM is to map the original data x into a feature space with high dimensionality through a non-linear mapping function and construct an optimal hyper-plane in new space. SVM models were constructed using different kernel functions. In order to study the performance of each kernel in predicting damage level of non-reshaped berm breakwater, SVM is trained by applying these kernel functions. Performance of SVM is based on the best setting of SVM and kernel parameters. Correlation Coefficient (CC) of SVM (polynomial) model (CC Train = 0.908 and CC Test = 0.888) is considerably better than other SVM models.

To avoid over-fitting or under-fitting of the SVM model due to the improper selection of SVM and kernel parameters and also the performance of SVM, hybrid particle swarm optimization tuned support vector machine regression (PSO-SVM) model is developed to predict damage level of non-reshaped berm breakwater. The performance of the PSO-SVM models in the prediction of damage level is compared with the measured values using statistical measures, such as, CC, Root mean Square Error (RMSE) and Scatter

Index (SI). PSO-SVM model with polynomial kernel function gives realistic prediction when compared with the observed values (CC Train = 0.932, CC Test = 0.921). It is observed that the PSO-SVM models yield higher CCs as compared to that of SVM models.

However, it is noticed that ANN model in isolation cannot capture all data patterns easily. Adaptive Neuro-Fuzzy Inference System (ANFIS) uses hybrid learning algorithm, which is more effective than the pure gradient decent approach used in ANN. ANFIS models were developed with different membership namely Triangular-shaped built-in membership function (TRIMF), Trapezoidal-shaped built-in membership function (TRAPMF), Generalized bell-shaped built-in membership function (GBELLMF), and Gaussian curve built-in membership function (GAUSSMF) to predict damage level of non-reshaped berm breakwater. The performance of the ANFIS models in the prediction of damage level is compared with the measured values using statistical measures, such as, CC, RMSE and SI. ANFIS model with GAUSSMF gave realistic prediction when compared with the observed values (CC Train = 0.997, CC Test = 0.938). It is observed that the ANFIS models yield higher CCs as compared to that of ANN models.

The different soft computing models namely, ANN, SVM, PSO-SVM and ANFIS results are compared in terms of CC, RMSE, SI and computational time. The hybrid models in both (ANFIS and PSO-SVM) cases showed better results compared to individual models (ANN and SVM). When the hybrid models are compared, ANFIS model gives higher CC and lower RMSE. But considering computational time, ANFIS has taken more time than PSO-SVM model. Hence PSO-SVM is computationally efficient as compared to ANFIS. ANFIS and PSO-SVM models perform better and similar to observed values. Hence, ANFIS or PSO-SVM can replace the ANN, SVM for damage level prediction of non-reshaped berm breakwater. ANFIS or PSO-SVM can be utilized to provide a fast and reliable solution in prediction of the damage level prediction of non-reshaped berm breakwater, thereby making ANFIS or PSO-SVM as an alternate approach to map the wave structure interactions of berm breakwater.

Keywords: Berm Breakwaters, Damage Level, Prediction, ANN, ANFIS, SVM, PSO-SVM.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PLATES

LIST OF NOTATIONS

Symbol	Description
A	Area of erosion
B	Berm width
d	Water depth above the bed level
Dn50	nominal diameter of the stones
h_B	Berm position above seabed,
H_0	Deep water wave height
L_0	Deep water wave length
h_B/d	relative berm position
K_D	Stability coefficient
M50	Median stone mass
S	Damage Level
W	Weight of the individual armor
W_{50}/W_{max50}	Armor stone weight
B/L_o	Relative berm width
h_B/L_o	Relative berm location
H_o/L_o	Deepwater wave steepness
N	number of waves
nsv	Number of support vector
Greek	
α	angle of the breakwater slope
ε	Error in error tube
ξ_μ	surf similarity parameter

γ	Specific weight of water
γ_a	density of stone
γ_ρ	specific weight of armor units
σ	Standard deviation
Δ	relative mass density of armor

Abbreviations

ANN	Artificial Neural Network
ANFIS	Adaptive neuro-fuzzy Inference system
CC	Correlation coefficient
erbf	Exponential radial basis function
FIS	Fuzzy Inference System
GP	Genetic programming
GA	Genetic algorithm
GA-SVMR	Genetic algorithm based support vector machine regression
NITK	National Institute of Technology Karnataka
PCA	Principal component analysis
PSO	Particle Swarm Optimization
PSO-SVM	Particle Swarm Optimization based Support vector machines
RMSE	Root mean square error
rbf	Radial basis function
SI	Scatter index
SWL	Still Water Level
SVM	Support vector machines

CHAPTER 1

INTRODUCTION

1.1 GENERAL

India is one of the world's fast emerging new economies. The development of economy is mainly possible by new infrastructure to connect its huge hinterland, so goods, commodities and the final products can be transported from the coast to India's interior and vice versa. This implies that harborage and harbor facilities should be improved. Further, with the development in coastal regions, the environmental stress is rapidly growing on the coastal belt which has to be protected and it is a challenge for the coastal engineers to protect the environment. The coastal structures built to protect the harbor and shores such as seawalls, groins, offshore breakwaters, artificial nourishments are required to withstand the destructive forces of the sea waves. These structures have been tried to overcome the problem of erosion and also for maintaining tranquility condition inside port and harbor for loading cargo and passengers. Some of them have been successful while some others have failed to achieve the job assigned to them. The failure may be due to improper location and design or wrong choice of protective measures. The cause for the erosion is generally due to the concentration of wave energy at a specific location. Hence there is a need to dissipate the wave energy before it reaches coast. The use of breakwaters is one of the solutions to dissipate the wave energy.

1.2 BREAKWATER

Breakwaters are widely used throughout the world for protection from the wave action and to provide shelter to harbor and port. They are used to protect beaches, cultivation land, and valuable habitats from erosion. Breakwaters are also used for dual purposes like dissipating wave energy and providing loading facilities for cargo and passengers. The alignment of the breakwater must be carefully considered after examining the predominant direction of approach of waves and winds, degree of protection required, magnitude and direction of littoral drift and the possible effect of these breakwaters on the shoreline. In general these studies are invariably conducted

in a physical model test where various alternatives are studied and the final selection will be based on performance consistent with cost. From economic point of view breakwaters represent a significant portion of capital investment in the development of port and would require a regular maintenance to retain their effectiveness.

Breakwaters are classified mainly as:

- Rubble mound or heap breakwaters,
- Upright or vertical wall breakwaters,
- Mound with superstructure or composite breakwaters,
- Special type of breakwaters.

1.2.1 Rubble Mound Breakwaters

These are trapezoidal shaped breakwaters and the oldest form of coastal protection work. A rubble mound breakwater dissipates the major part of the incoming wave energy by inducing wave breaking on the slope and partly by porous flow in the mound. The remaining energy is partly reflected back to the sea and partly transmitted into the harbour area by wave penetration and wave overtopping (if the breakwater is low crested). Various kinds of rubble mound breakwaters have been constructed depending on the purpose of the breakwater (Fig. 1.1).

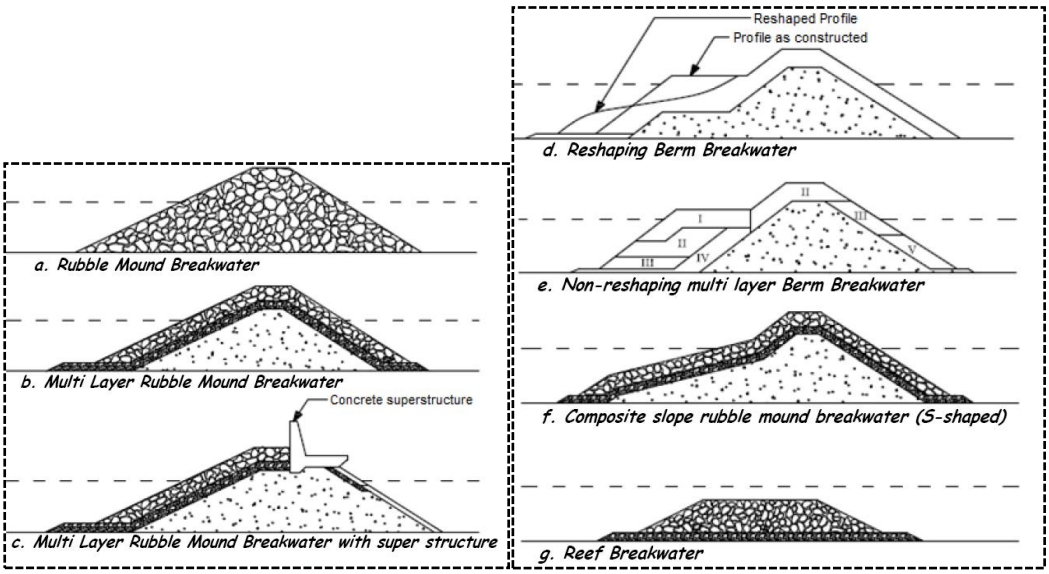


Fig. 1.1 Types of rubble mound breakwaters (Andersen, 2006)

The most simple breakwater consist only of a mound of stones called as rubble mound breakwater (Fig. 1.1-a). However, this type of structure is very permeable, and may

cause significant wave and sediment penetration. In addition, large stones are expensive, because most quarries generate a lot of finer material and a relatively small number of large stones. Fig. 1.1-b and 1.1-c are the two most common types of rubble mound breakwaters known as conventional rubble mound breakwaters with and without a superstructure respectively. These consist of various layers with primary layer exposed to wave action. The primary armour layer consists of rock or concrete units which are large and heavy to remain in their position during wave conditions. This layer also protects the bottom layers namely secondary layer and core. The secondary layer prevents finer material being washed out through the primary armour layer acts as filter layer. The core consists of fine material which acts as barrier for the water to pass through it. A superstructure is provided when overtopping is not to be allowed.

Since the 1980s a design based on natural reshaping of the front rock armour during wave action has gained more attention (Fig. 1.1-d) (Andersen, 2006). This type of breakwater is known as berm breakwaters or reshaping breakwaters. The main advantage of this structure is that simpler construction methods can be applied. The berm breakwater concept is described in detail in later sections.

Lately, non-reshaping berm breakwaters have been considered, often with several stone classes to maximize the total stability and quarry utilization, as indicated in Fig. 1.1-e. Especially in Iceland this structure is widely used, and is therefore also known as the Icelandic type of berm breakwater. Structures constructed with an S-shaped profile (Fig. 1.1-f) are typically used in large water depths to reduce the volume of material, but construction costs are in most cases significantly larger than for the reshaping berm breakwater, resulting in approximately the same profile. Fig. 1.1-g shows a reef breakwater which is a submerged breakwater mainly used for protecting beaches.

1.2.2 Upright or Vertical Wall Breakwaters

Vertical or upright wall breakwaters (Fig. 1.2) are of huge concrete blocks, gravity walls, concrete caissons, rock filled timber cribs and concrete or steel sheet pile walls. The selection of type of breakwater would be primarily based on the wave climate in that area, depth of water, availability of construction materials and local manpower,

geotechnical nature of seabed, function of breakwater, technical know-how and contractor potential available.

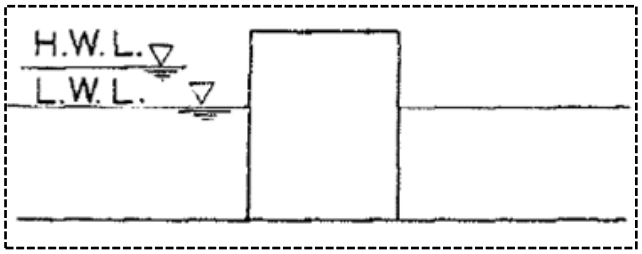


Fig. 1.2 Vertical wall breakwater (Takahashi, 1996)

1.2.3 Mound with Superstructure or Composite Breakwaters

Composite breakwaters are combination of rubble mound and vertical wall. These are used in locations where either the depth of water is large or there is a large tidal range and in such situations, the quantity of rubble stone required to construct a breakwater to the full height would be too large. In such conditions, a composite breakwater is constructed which is a structure with rubble mound base and a super structure of vertical wall as shown in Fig 1.3.

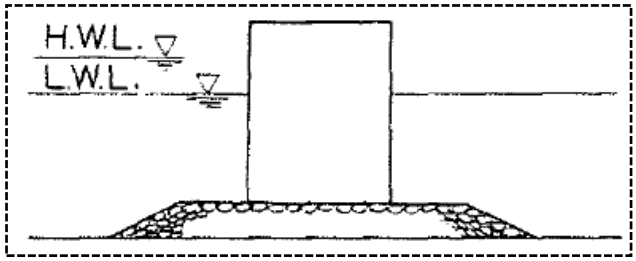


Fig. 1.3 Composite breakwater (Takahashi, 1996)

1.2.4 Special Type of Breakwaters

Special type breakwaters are those employing some kind of special features and are not commonly used. Special type breakwaters can be divided into two kinds. One is the non gravity type breakwaters such as a pile type, floating, pneumatic etc. The other is the conventional breakwater with special features conceived to improve the function and stability of breakwater. Fig. 1.4 shows some of the special types of breakwaters. Some special breakwaters are as follows;

a) Curtain wall breakwater – commonly used as secondary breakwater to protect small craft harbours.
b) Sheet pile walls – used to break relatively small waves.

c) Horizontal plate breakwater – can reflect and break waves and are supported by a steel jacket.

d) Floating breakwater – very useful as breakwater in deep waters especially in places where the ground soil is poor for foundation.

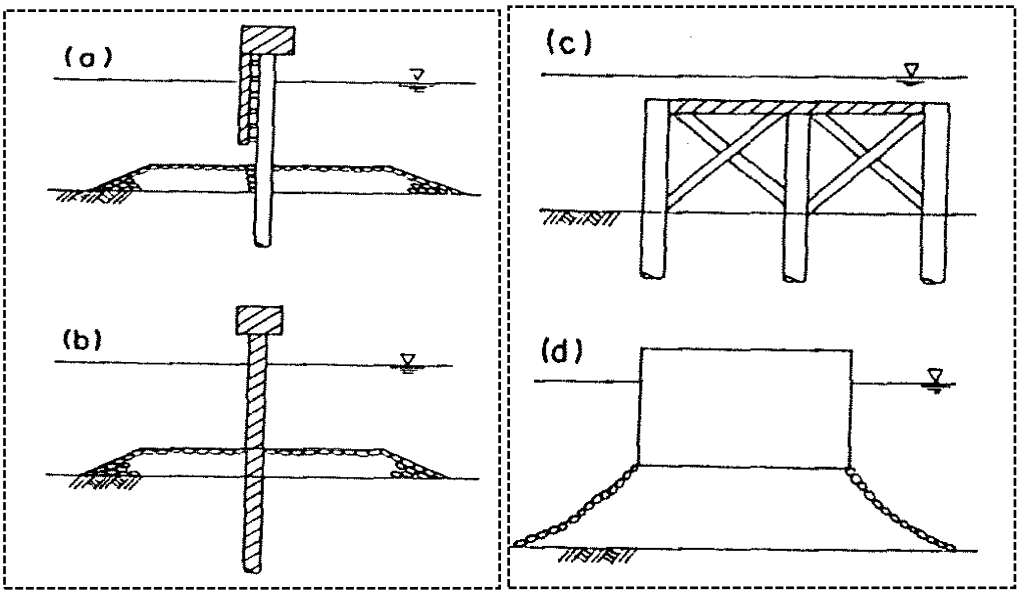


Fig. 1.4 Special breakwaters (Takahashi, 1996)

1.3 BERM BREAKWATER

A berm breakwater is a rubble mound breakwater with a horizontal berm above or at still water level (SWL) on the seaward side (Fig 1.5). During exposure to wave action of certain intensity, the berm reshapes until eventually an equilibrium profile of the stones on the seaward face is reached. The basic principle involved in the concept of berm breakwaters is the utilization of smaller stones available locally. The dimension of the horizontal berm at or around the water level is to be determined by model studies. The berm breakwater will not fail in a catastrophic manner because the permeable berm of smaller stones consolidates into a nested surface. The relatively high porosity of the berm allows the waves to propagate into the armor stones and dissipate the energy over a large area (Baird and Hall, 1984).

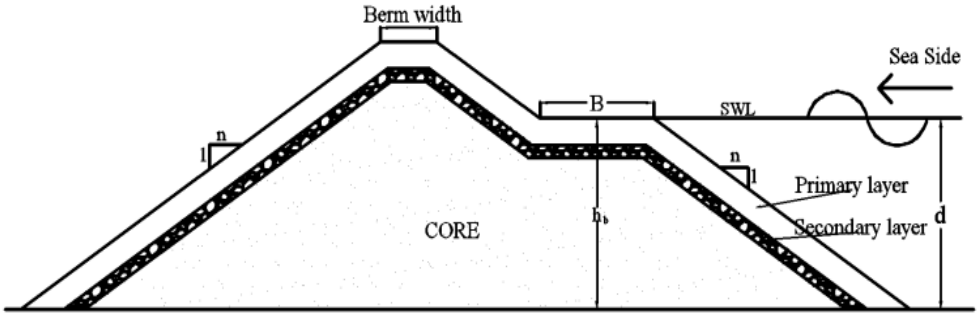


Fig. 1.5 Berm breakwater

Berm breakwater has a main feature consisting of rather thick cover layer of stones, relatively much smaller than on a conventional breakwater which comprises of one or two layers of cover blocks. The berm breakwater has been adopted at several places as an economic solution when large cover blocks of natural stones are not available. It might also be an economical solution even when large cover blocks are available for conventional breakwater. The uncertainty in the wave climate favors a breakwater design that is not too sensitive to the wave height with respect to stability.

Berm breakwaters are further classified by PIANC (2003) based on its reshaping as:

- **Statically Stable Non- Reshaped**: These are similar to conventional breakwaters with only few stones allowed to move. The H/ΔD (Stability number) value is less than 1.5 these structures.
- **Statically Stable Reshaped**: The profile of the breakwater is allowed to reshape to an equilibrium profile after which the stones are also stable without any movement. For these structures, H/ΔD value ranges from 1.5-2.7.
- **Dynamically Stable Reshaped**: The profiles in these structures attain an equilibrium profile, but, the individual stones may move up and down the slope. The H/ΔD value is more than 2.7 for dynamically stable reshaped berm breakwaters.

The advantages of berm breakwater are:

- Use of smaller stones that are easily available at the site
- Weight of stones can be 20% to 30% of the armor units required in conventional design

- Optimum utilization of quarry yield
- Simple design and construction method
- Enhanced speed of construction
- Though volume may be more compared to conventional breakwater, effective cost savings due to ease of construction utilizing smaller cranes etc.

1.4 DAMAGE LEVEL

The amount of displacement of rock to be expected in the structures life time and under design conditions, is an essential parameter in the design process of statically stable rubble mound breakwater. This amount of displacement is called damage. A meaningful definition for 'damage' is essential for qualitative analysis of stability of rubble mound structures. The traditional method of quantifying damage was physical counting of displaced units caused by wave action. The damage parameter was defined by Hudson (1959) as the percentage of armor units displaced from the cover layer. The removal of up to 1% of the total number of armor units in the cover layer is considered as 'no damage'.

According to Van der Meer (1988) the term "Damage" is defined as the displacement of armor units (Fig 1.6). Relation used to measure the damage level (S) is

$$S = \left({}^{A}/_{D_{n50}{}^{2}} \right) \tag{1.1}$$

$$D_{n50} = \left({}^{M_{50}}/_{\rho_a} \right)^{1/3} \tag{1.2}$$

where, A is the area of erosion and D_{n50} is the nominal diameter of the stones M_{50}= median stone mass, and ρ_a=density of stone. A physical description of the damage S was given as the number of squares with side D_{n50} that fitted into the erosion area. Another description was that the number of cubic stones with a side of D_{n50} eroded within a width of one D_{n50}. The actual number of stones eroded within this width of one D_{n50} could be more or less than S, depending on the porosity, the grading of the armor units and the shape of the stones. But generally, the actual number of stones eroded within a width of one D_{n50} was equal to 0.7 to 1.0 times the damage S.

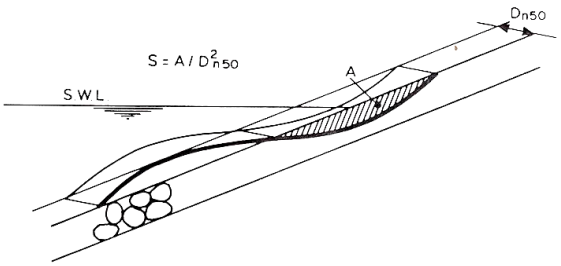


Fig. 1.6 Erosion area and damage level, S (Van der Meer 1988)

CEM (2006) gives a conventional damage level classification and related values of the damage parameters, S as shown in Table 1.1.

Table 1.1 Damage classification and related values of damage parameter

No damage	No unit displacement.
Initial damage	A few units are displaced. This damage level corresponds to the no-damage level used in *Shore Protection Manual* 1977 and 1984 in relation to the Hudson formula stability coefficient, where the *no-damage* level is defined as 0-5% displaced units within the zone extending from the middle of the crest height down the seaward face to a depth below SWL equal to a H_s-value which causes the damage 0-5%
Intermediate damage ranging from moderate to severe damage	Units are displaced, but without causing exposure of the under or filter layer to direct wave attack, S = 5-10% for rock slopes 1:2 -1:3
Failure	The under layer or filter layer is exposed to direct wave attack, $S \geq 20\%$

1.5 SCOPE OF THE PRESENT INVESTIGATIONS

In early decades, many researchers carried out experimental, theoretical and numerical investigations on breakwaters (Priest et al., 1964; Brunn et al., 1976; Baird and Hall, 1984; Ergin et al., 1989; Sigurdarson et al. 1998; Hall and Kao, 1991; Van der Meer, 1992 and 1996; Torum et al., 1999 and 2003; PIANC, 2003). Physical model studies on breakwaters involve several common assumptions applied in hydrodynamics which may not be accurate.

In physical modeling, there are number of uncertainties associated with many of the governing variables and their effects on the performance of breakwater. Engineering judgments based on experience, subjectivity, confidence on model, and other factors are frequently used to deal with this non-statistical uncertainty. Most of the problems need to consider a large number of parameters and dimension, which affect the response of the systems. Thus, complexity is also an inherent feature of these problems. Because of these characteristics, conventional mathematical modeling for predicting the performance of breakwaters tends to become very difficult and often the prediction is quite unreliable. Further, a simple mathematical model is not available to predict the damage of berm breakwaters because of these complexities.

From the literature review (Chapter 2), it is found that, soft computing techniques, such as, Artificial Neural Network (ANN), Fuzzy Logic (FL), Adaptive Neuro Fuzzy Inference System (ANFIS), Support Vector Machine (SVM), Particle Swarm Optimization (PSO) etc., are successfully applied to coastal engineering problems which are complex in nature. However, it is observed that there are hardly any applications of soft computing techniques to predict damage level of non-reshaped berm breakwater.

1.6 ORGANIZATION OF THE THESIS

The thesis is presented in six chapters:

Chapter 1 - Introduction: Introduction to breakwaters, classification of breakwaters and its damage level and scope of the present investigations has been discussed.

Chapter 2 - Literature review: The literature review on theoretical and analytical studies specifically related to berm breakwaters, applications of soft computing techniques in coastal/ocean engineering, problem formulation and objectives of present work have been discussed.

Chapter 3 – Experimental Data and Methodology: Briefly explained features of experimental model setup, experimental investigations carried out in the wave flume and data used for developing soft computing models. Also, theoretical background of research methods used to developed soft computing models to predict damage level of

non-reshaped berm breakwater, such as, ANN, SVM, ANFIS and PSO has been discussed.

Chapter 4 – Results and Discussion: The results obtained from the soft computing models, such as, ANN, SVM, PSO-SVM and ANFIS in prediction of damage level of non-reshaped berm breakwater are analyzed, interpreted and discussed. Also, the performance of these models is compared with each other.

Chapter 5 – Summary and Conclusions: Brief summary of the research work and the conclusions drawn based on the results of soft computing models and suggestions for future work have been presented.

Appendix I includes the MATLAB programs used to develop various soft computing models.

The Appendix I is followed by references, list of publications based on the present work, and a brief resume of the researcher.

CHAPTER 2

LITERATURE REVIEW

2.1GENERAL

Berm breakwaters were introduced in early 1980's. At large water depths, where design waves are high, berm breakwater of two layer armor stone have proved to be economical. Layers can be constructed and designed using smaller stones to fit the available armor stone in quarry. The wave energy and hydrodynamic force is dissipated in the armor stone mass. Due to this force smaller stones get displaced within the layer and results in the formation of a stable layer.

In order to understand the behavior of berm breakwater due to wave action a detailed study on berm breakwater is necessary. It is important to get safe and optimal structure. The studies can be carried out as a physical model study or analytical study. The advantage of analytical study over physical model study is explained in section 1.2 of Chapter 1. The techniques such as artificial neural networks, fuzzy logic, particle swarm optimization, genetic algorithms, support vector machines or combinations of these soft computing techniques can be adopted to carry out studies on berm breakwaters.

Use of soft computing techniques can be an effective alternative due to following reasons:

1. Considering the coastal boundary and depth variation, field analysis of wave structure interaction, determination of stability and damage level of berm breakwater structure is difficult.
2. Mathematical modeling of these complex interactions is difficult while physical modeling will be costly and time consuming.

In this chapter, a detailed review of literature on berm breakwaters, soft computing tools used in coastal engineering field which is carried out to know the advancement in research and the advantages or drawbacks of the available methods.

2.2 REVIEW OF STUDIES ON BREAKWATER

Many researchers have carried out experimental and numerical studies on breakwaters in the past and some of them are discussed below:

Hudson (1959) conducted laboratory investigations to determine the criteria for the design and construction of rubble-mound breakwaters. A general stability equation was derived which was used to guide the experimental and test data. A new breakwater stability formula was obtained after unknown function in the stability equation was determined for the selected breakwaters and test wave conditions by using the test data results. Also wave run-up studies were carried out. According to Hudson, the weight of the individual armor unit is given by

$$W = \frac{\gamma_r H^3}{K_D \Delta^3 \cot \alpha} \tag{2.1}$$

Where, γ_r is the specific weight of armor units; H is the design wave height; Δ is the relative mass density of armor and K_D is the stability coefficient. In deriving the stability formula Hudson neglected the friction between the armor units. It was also assumed that the dynamic effect of the waves was to lift and roll the armor units from their initial positions on the breakwater slope.

A mathematical model was developed by Kobayashi and Jacobs (1985) to predict the flow characteristics in the down rush of regular waves and movement of armor units on the slope of a coastal structure. Stability analysis of armor units was performed including the drag, lift and inertia forces acting on an armor unit which vary along the slope. Then the comparison was made with the large scale test data based on their earlier study. Their model did not consider the effects of permeability, bottom friction and water depth, but, the model was able to predict the observed zero-damage stability number satisfactorily.

A mixed numerical model was developed by Hannoura and McCorquodale (1985) to simulate wave motion in rubble-mound structures. Their model was applied to check the dynamic stability of the seaward slope under severe wave attack of the sines breakwater. The predicted internal water surface values were compared with a

physical model measurement which was found to be reasonable. But, the model showed a lower factor of safety than the traditional analysis. They provided additional terms to take into account the effect of added mass which improved the detection of internal wave breaking and the entrainment of air near the interface.

Van der Meer (1988) established new stability formulae for rubble mound breakwaters under random wave attack with comprehensive model investigations at Delft Hydraulics, Netherlands. The experiments were conducted to include variety of structures with differing core and under layer permeability for a wide range of wave conditions. Considering type of wave breaking two formulae were derived, one for plunging waves and other for surging waves. These equations are now popularly known as the Van der Meer formulae. All the test results showed a clear difference between plunging and surging waves. The minimum stability of a structure was during the transition of waves from surging to plunging waves. Using curve-fitting technique, Van der Meer (1988, 1988a) specified two stability formulae, as follows.

For plunging waves:

$$\frac{H_S}{\Delta D_{n50}} = 6.2P^{0.18}\left(\frac{S}{\sqrt{N}}\right)^{0.2}\xi_m^{-0.5} \tag{2.2}$$

For surging waves:

$$\frac{H_s}{\Delta D_{n50}} = 1.0\,P^{-0.13}\left(\frac{S}{\sqrt{N}}\right)^{0.2}(\cot\alpha)^{0.5}\xi_m^{P} \tag{2.3}$$

where, Δ is the relative mass density of armor, S is the damage level, H_s is the significant wave height at the toe of the structure in meters, α is the angle of the breakwater slope with horizontal, P is the permeability coefficient, N is the number of waves, ξ_m is the surf similarity parameter using average wave period T_m, D_{n50} is the nominal diameter of the armor unit in meters.

Another equation to calculate the critical ξ_m during the transition from plunging to surging waves was given as,

$$\xi_m = \left(6.2P^{0.31}\sqrt{\tan\alpha}\right)^{1/(P+0.5)} \tag{2.4}$$

Depending on slope angle and permeability the transition lays between ξ_m = 2.5 to 4. For cotα greater than or equal to 4.0 the transition from plunging to surging does not exist and for these slope angles equation (2.3) must be used.

The wave structure interaction of berm breakwaters was studied by Van Gent (1995) with the help of physical as well as numerical models. Wave motion calculated by the numerical model was verified by physical model tests. The numerical study considering the finite-amplitude shallow-water wave equations was able to simulate the wave motion both on and inside the structure. He included a new morphological model for cross-structure transport into the initial model for incident waves which resulted in a new combined wave load-response model that was capable of simulating the reshaping process of coastal structures like berm breakwaters and gravel beaches. The combined numerical model was compared with physical model test data and also with prototype data. From the comparison he concluded that the developed numerical model was able to predict the influence of parameters such as wave height, wave period, and stone diameter on the reshaped seaward slopes fairly accurately.

Tørum (1988) carried out laboratory tests on berm breakwaters in shallow water, e.g. in water depths where waves might break before they break on the breakwater. Analysis was made on test results for breakwaters in shallow and deep waters. Combined equations for the berm recession of berm breakwaters in shallow and deep water were presented.

Melby and Kobayashi (1998) carried out an experiment on breakwater damage by considering long duration using irregular waves in a flume. By varying the parameters like wave height, wave period, water depth, storm duration, and stone gradation systematically, a new damage measurement technique was developed and damage development data was acquired for breaking wave conditions. Their experiment yielded relationships for both temporal and spatial damage development. The equations for predicting temporal variations of mean damage with wave height and period varying with time in steps, developed by Melby and Kobayashi (1998a, b), performed well in describing the damage of the breakwater. For new test series, the

damage initiation was consistently predicted by more than a standard deviation. The prediction was shown to improve significantly if the initial profile adjustment was accounted in the test series with relatively small cumulative damage.

Experiments were conducted by Tørum et al. (2003) to study the stability of multilayer berm breakwaters. The multilayer was achieved by using two different mass densities of stone. An equation for berm recession was derived considering stone gradation and water depth. They also investigated the erosion in berm breakwaters and wave overtopping discharges. From their research they concluded that the stone size required for toe protection could be calculated using the existing formulae, but, only as a first approximation. They observed that overtopping was less in case of reshaped berm breakwaters when compared to conventional rubble mound breakwaters. They also noticed that scour was greatly affected by scale of the tests and it was found to be increasing with larger scales.

Melby and Hughes (2003) developed an equation considering the wave momentum flux which was found to be proportional to the maximum wave forces on armor units. They also developed two equations, each, for surging and plunging waves as developed by Van der Meer (1988) considering his small scale test data. Their data was limited mostly for non-breaking waves, normally incident waves, and near-shore slope 1V:20H, non-overtopped structures and angular randomly-placed armor stone. Wave-structure interaction and near-shore nonlinear wave transformation were separated out to overcome a major limitation of pervious stability relations. The newly developed relations clearly illustrated the influence of water depth and wave period on stability.

Rao et al. (2004) analyzed experimentally the stability of S-shaped breakwater with reduced armor unit weight and the influence of wave height, wave period, and berm width on the breakwater. The studies showed that 30% reduced armor weight model with 0.6 m berm was stable for the design wave height. Their results showed a large influence of wave period on the stability of berm breakwaters.

The influence of wave height, wave period, water depth and seaward slope on the stability, wave run-up, and wave run-down of statically stable rubble-mound berm breakwater was investigated by Rao et al. (2008) through physical model studies. They observed a greater influence of wave height and wave period on the stability of berm breakwaters. They found gentle slopes of breakwater prone to less damage compared to steeper slopes and even wave run-up and run-down were less on gentle slopes.

Losada et al. (2008) described the results of a two-dimensional numerical modeling investigation conducted on rubble mound breakwaters mainly focusing wave overtopping process. They adopted a new model, COBRAS-UC, which was an updated model of COBRAS (Cornell Breaking Waves and Structures).They used Volume of Fluid Technique (VOF) to capture the free surface. Several phenomena like wave run-up, wave reflection, wave transmission through rubble mounds, wave overtopping and agitation at the lee side were simulated using the model. They also carried out two-dimensional experimental studies to verify the performance of model. Prediction by the model was in good agreement with the experimental results.

A probabilistic hydrodynamic model for the prediction irregular wave action on the wet and dry zone of a permeable structure was developed by Kobayashi et al. (2010). The results of the four test series was used for comparison with the model. The comparison test results showed that water depth, velocity, and discharge exceeded by 2% of 1000 wave incidents. Input from the hydrodynamic model was used to modify a formula for bed load on beaches for predicting the damage progression of a stone armor layer. The eroded area of the damage layer was predicted well by the modified model, but, the deposited area was over estimated. This was because of not considering the discrete stone units deposited at the toe of the structure.

Van Gent (2013) developed an empirical equation for stability of rubble mound breakwaters with berm armored with rock through experimental study. The tests mainly focused on the slope above and below the berm, and also the stability of the rock at the berm. The influence of the level of the berm, slope angle, wave steepness

and width of the berm were also investigated. From the study, the rock size on the upper slope of the berm could be reduced significantly compared to straight slopes.

Many researchers through their studies have proved that armor stability is a function of wave height, wave period, wave groupiness, wave breaking, water depth, structure slope, armor unit weight, armor gradation, storm duration and porosity. Each of them investigated structure stability with respect to different parameters (Hudson 1959; Ahrens 1970; Ergin and Pora 1971; Bruun and Gunbak 1976; Kondo et al. 1976; Johnson et al. 1978; Carver and Davidson 1982; Ahrens 1984; Timco et al. 1984; Van der Meer and Pilarczyk 1984; Gadre et al. 1985; Van der Meer 1988; Hall and Kao 1991; Poonawala et al. 1994; Hegde and Samaga 1996; Rao 2000; Rao et al. 2004 and 2008).

But from the review literatures, it is found that the theoretical determination of stability and damage of breakwaters is difficult because of the difficulty in modeling the energy dissipation at the structure which is basically due to the turbulence caused at the structure. Moreover, there is no mathematical model to determine the damage level of breakwaters. Therefore, it is necessary for researchers to adopt the physical model study to quantitatively determine the parameters that influence the phenomenon. However, there are some difficulties in physical modeling that makes it inconvenient to use. Firstly, physical modeling is a time consuming process and expensive. It requires costly set-up and continuous maintenance. The experimental facility and instruments needs to be calibrated regularly before usage, failing as such may cause experimental errors. Further, area required for setting up of the model study is large and needs skilled men for operating the instruments. Alternatively, soft computing techniques can overcome these disadvantages and also provide fast and reliable solution in predicting the damage level of berm breakwaters. The next few paragraphs highlight the usage of these soft computing techniques in coastal field.

2.3 REVIEW OF LITERATURE ON APPLICATIONS OF SOFT COMPUTING TECHNIQUES IN COASTAL / OCEAN ENGINEERING

Several researchers have adopted soft computing techniques to solve complex associated with coastal/ocean engineering problems and some of them are discussed below:

Neural networks technique was applied to predict the stability of rubble mound breakwater (Mase et al., 1995). They considered several parameters like stability number, permeability of breakwater, damage level, surf similarity parameter, number of attacking waves, spectral shape parameters and dimensionless water depth. The developed model predicted the damage level satisfactorily.

Mase and Kitano (1999) applied Neural Network (NN) techniques to predict impact wave force which would act on the upright section of a composite breakwater. Four non-dimensional parameters were fed as input *viz.* h/L, H/h, d/h, and B_m/h where, h denotes the total water depth, H represents the wave height, L is the wavelength, B_m is the horizontal distance from the shoulder of mound to the caisson and d is the water depth above the mound. The parameters of the model were determined through self-learning and the results were accurate.

Deo and Kumar (2000) estimated the weekly mean significant wave heights using neural network, and model based statistical and numerical methods, from their monthly mean observations. They trained the network using error back propagation, conjugate gradient and cascade correlation algorithms. The training of the model using cascade correlation took minimum training time with better Correlation Coefficient (CC) between observations and network output.

A model to estimate the significant wave heights and average wave period from the generating wind speed was developed by Deo et al. (2001) using 3-layered feed forward neural network. They trained the network with different algorithm and used three sets of data. The trained network provided satisfactory results in deep water, in

open wider areas and also when the sampling and prediction interval were large, such as a week. They also revealed that a proper choice of training patterns was found to be crucial in achieving adequate training.

The back propagation neural network method was applied for accurate prediction of tides (Mandal et al., 2001). Their neural network model predicted the time series of hourly tides using quick learning process called quick prop. The CC between predicted tides and measured tides was found to be 0.998. It showed a good agreement between neural network prediction and measured data set.

Large data requirement by the traditional method for prediction tidal level can be avoided by adopting NN with little dataset (Lee and Jeng, 2002).

To predict long term water levels in a coastal inlet a Regional Neural Network–Water Level (RNN-WL) model using feed forward, back propagation neural network structure was developed by Huang et al. (2003). Hourly data of over one month period was used for training the network and another one month for validating the model. The model was then tested for prediction over year long periods. The model predicted the long term tidal as well as non-tidal water levels in the coastal inlets fairly accurate even though there were significant changes in the amplitude and phases of the water levels over the regional study area. The model was then updated to examine the effect of distance on the model performance. The results of the model indicated that RNN-WL model could be used as a supplement for the long-term historical water level data.

Deo and Jagdale (2003) developed NN model for prediction of breaking waves. They trained the network by using the existing deterministic relations with a random component. Fresh laboratory observations were used for validating the model. The results showed that the prediction of breaking height and water depth by the model were more accurate than the available traditional empirical formulae.

An ANN model to improve wave short term forecasts was developed by Makarynskyy (2004). The model was trained and validated using hourly observations

of significant wave heights and zero-up-crossing wave periods from two offshore sites. Two approaches were adopted for the forecasting. One approach corrected the predictions using the initial simulations of the wave parameters with lead times from 1h to 24h and the other approach was used for merging the measurements and initial forecast. Results showed satisfactory predictions at both locations.

Lee (2004) developed a back propagation neural network model using short term on site tidal level data obtained from Taichung Harbor in Taiwan. Model predicted results were compared with conventional harmonic methods, indicates that the back propagation neural network efficiently predicted the long term tidal levels.

The feed forward neural network was used to predict hourly sea level variations for 1/2, 1, 5 and 10 days mean sea level (Makarynskyy et al., 2004). The results showed that the sea level prediction was good in terms of CC (0.7-0.9), Root Mean Square Error (RMSE) (about 10% of tidal range) and Scatter Index (SI) (0.1-0.2).

Mandal et al. (2005) estimated various ocean wave parameters using theoretical Pierson-Morskowitz (PM) spectra as well as measured ocean wave spectra using back propagation neural network. According to them, very high CC for significant wave height (H_s), maximum spectral energy (E_{max}), zero crossing wave period (T_z) and time period for maximum spectral energy (T_p) in training the NN for PM spectra due to Gaussian distribution justified the use of NN.

The ANN approach was used for estimating the wave parameters from cyclone generated wind fields (Rao et al., 2005). Estimation of Hs and periods was carried out using back propagation neural network with three updated algorithms, namely Rprop, Quick prop and superSAB. The predicted values using neural networks matched well with those estimated using Young's model and a high CC of 0.99 was obtained.

The feed forward back propagation neural network was used to predict the significant wave height (Hs) values sensed by a satellite at deeper locations (Karla et al., 2005).

The NNs provide a useful tool to predict deep-water waves sensed by satellite to coastal locations.

The ANN technique was used to predict the significant wave height (Hs) and zero crossing wave periods (Tz) (Makarynskyy et al., 2005). They achieved a higher accuracy of simulating the Hs and forecasting Tz by using ANN.

A NN was developed in order to estimate the wave surface density over a wide range of wave frequencies from average wave parameters of Hs, Tz, spectral width and peakedness parameters (Naithani and Deo, 2005). They compared the NN predicted values with the measured ones. The predictions were acceptable than those yielded by PM, JONSWAP and Scott's spectra.

The Adaptive Network based Fuzzy Inference System (ANFIS) and Coastal Engineering Manual (CEM) methods were used to predict the ocean wave parameters (Kazeminezhad et al., 2005). According to them, the results indicated that the ANFIS outperforms CEM method in terms of prediction capability. Here the CEM method overestimated the Hs and underestimated the peak spectral period, while ANFIS resulted in predictions that are more accurate.

The functional and sequential learning neural networks were applied for accurate prediction of tides using very short-term observations (Rajasekaran et al., 2005 and 2006). The comparison between the measured and predicted tidal levels for 3 days and 1 month's prediction using 1 day observation showed the CCs, 0.981 and 0.999, which were higher than the values obtained by Tsai and Lee (1999). It showed that the functional and sequential learning neural networks predicted better values as compared to other conventional methods.

Yagci et al. (2005) used NN technique to predict the damage ratio of breakwater. According to them, the accurate estimation of damage levels of breakwater was vital issue in design of breakwater. The network was constructed by considering input parameters like wave stiffness (Hs/Ls), significant wave period (Ts) and slope angle (α).

They have used fuzzy logy system for mapping the inputs and output. The fuzzy model estimations of damage ratios were close to the predicted values by NN methods. The employment of Artificial intelligence (AI) methods enables the consideration of wave period; wave stiffness, breakwater slope and wave height in estimating damage ratio. This application is useful especially when there is less number of laboratory data set. The experimental data set were plotted effectively using AI technique in order to generate more number of data set.

The ANN was applied to design rubble mound breakwaters (Kim and Park, 2005). ANN technique yielded more accurate results compared to the conventional empirical model and also the accuracy of the results was influenced by NN structure. They showed that the solutions of Monte Carlo simulation technique could be improved by incorporating the trained neural network model into it.

Chang and Lin (2006) developed a new tide generating neural network model (TGF-NN) for simulation of tides at multi-points considering tide-generating forces. They compared the RMS and CC of many models for an input of three year mixed tides at a single point. They used with harmonic method, the NAO.99b model, response orthotide method and the TGF-NN model for the prediction. They found that the new TGF-NN model was more effective compared to the harmonic method. They also observed that the TGF-NN model could predict tides at some neighboring points more accurately compared to NAO.99b numerical model.

Perceptrons were trained using Particle Swarm Optimization (PSO) model by Chau (2006). The model was used for prediction of water levels in ShingMun river of Hong Kong. Different lead times calculated on the basis of the stage/time history or upstream gauging stations was adopted at the specific station. It was observed that PSO can be an alternative technique to ANN.

The recurrent neural network with updated algorithms was used to forecast ocean waves (Mandal and Prabaharan, 2006). The recurrent neural network of 3, 6 and 12 hourly wave forecasting yielded the correlation coefficient (CC) of 0.95, 0.90 and

0.87 respectively. According to them, the wave forecasting using recurrent neural network yielded better results compared to previous NN applications.

The marine structures in Taiwan suffer from typhoon attack every year. The earlier theoretical models were not properly predicting the typhoon waves. According to Chang and Chien, (2006), ANN-multi trend-simulating transfer function model accurately forecasted wave peak.

Karla and Deo (2007) trained the data in an innovative manner to address the wind speeds modeling problems where the wind speeds have a very high variation in their magnitude. The wind speeds and wave data collected by TOPEX satellite was used in radial basis function neural network. They found that combined network involving all the three parameters namely, wave period, wave height and wind speed would work more flexibly than the individual models. They also found that network training based on statistical homogeneity of data set was essential to obtain accurate results.

Browne et al. (2007) compared the traditional Spectral Wave Simulation Model (SWAN) and a model developed with two data driven approaches (linear and ANN) for estimating the waves near shore waves. They used the data collected from 17 near-shore locations for the assessment of their performance. The data gathered had heterogeneous geography and bathymetry and was recorded over a 7 month period around the Australia continent. They found that ANN's outperformed SWAN. Further, they also observed that the non-linear architecture consistently surpassed the linear model.

Mandal et al. (2007) used neural network technique to predict the stability number and damage levels of rubble mound breakwater. It was seen that a good correlation was obtained between network predicted stability numbers and estimated ones with less computational time as compared to Mase et al. (1995) and Kim and Park (2005).

Gent et al. (2007) developed a neural network model to estimate wave-overtopping discharges for wide range of coastal structures.

Bateni and Jeng (2007) developed ANFIS models for predicting scour depth as well as scour width for a group of piles supporting a pier. Two combinations of input data were used. One involved combination of non-dimensional parameters such as wave period, wave height, and water depth, while the other contained combinations of non-dimensional numbers which comprised of Reynolds number, the shields parameter, the Keulegan-Carpenter number, and the sediment number. The test results showed that ANFIS performed better than the existing empirical formulae. They also found that the ANFIS performed well with original (dimensional) parameters rather than non-dimensional data. It was also noticed that the depth of scour was more accurately predicted than its width.

Mandal et al., (2008) applied NN technique in predicting the stability number and compared it with the estimated stability number by Hudson and Van der Meer. The neural network was modeled with parameters, which affects the stability namely Permeability of breakwater, number of attacking waves, significant wave height, mean wave period, damage level, slope angle, berm width and reduced armor weight ratio. It was found that the network predicted lesser armor units as compared to empirical formulae which could make the design more economical and safe. The CC between the estimated stability number by empirical formulae and predicted stability number by neural networks were close to one.

Zamani et al. (2008) developed ANN and Instance-Based Learning (IBL) models to forecast significant wave heights for several hours ahead using buoy measurements. Experiments showed that the ANN's yielded slightly better agreement with the measured data than IBL. According to them, ANN's could also predict extreme wave conditions better than the other existing methods.

Gaur and Deo (2008) applied genetic programming (GP) to forecast ocean waves on real-time. They analyzed the wave rider buoy which was measurements at two locations in the Gulf of Mexico. The forecasts of significant wave height were made over lead times of 3,6,12 and 24h. They used a sample size belong to a period of 15

years and a testing period of 5 years. The forecast made by the approach of GP could be regarded as a promising tool for future applications to ocean predictions.

Londhe (2008) presented ANN and GP for estimation of missing wave heights at a particular location on a real time basis using wave height at other locations. Both approaches performed well in terms of accuracy of estimation, whereas GP model worked better in case of extreme events.

ANN and regression method was used by Gunaydin (2008) to predict the mean monthly significant wave height from meteorological data. Seven different ANN models were compared each comprising of various combinations of input. The inputs were monthly mean wind speeds, air temperature ratios and sea level pressures collected on hourly basis. Out of seven ANN models that the model with all the input parameters outperformed other models.

Beltrami (2008) incorporated an ANN algorithm in bottom pressure recorders (BPRs) software to improve its real time automatic detection of a tsunami within recorded signals. The performance and efficiency of ANN algorithms was compared to the one developed under the Deep-ocean Assessment and Reporting of Tsunami (DART) program run by the U.S. National Oceanic and Atmospheric Administration (NOAA). Results indicated that ANN algorithm showed an improvement in tsunami detection.

Liang et al., (2008) developed three back-propagation neural networks (BPNN) models viz; difference neural network model (DNN) for the supplementing of tidal record; minus-mean-value neural network (MMVNN) for corresponding prediction between tidal gauge station and weather-data based neural network model (WDNN) for set up and set down. They found that above models performed well in the prediction of tidal level or supplement of tidal record including strong meteorological effects.

For forecasting of wave parameters considering the wind speed and its direction, and the lagged wave characteristics, Takagi- Sugeno rule based fuzzy inference system

was used by Sylaios et al. (2009). The initial and final antecedent fuzzy membership functions were identified using Subtractive clustering method. The developed model was verified by predicting the wind and wave dataset recorded in years 2000 to 2006 (12,274 data points) in the Aegean Sea collected by an oceanographic buoy. For the training period, the model showed perfect fit and further was verified using the 2006 data (1,044 data points). Significant wave height and zero up-crossing periods for a lead time of three hours were predicted well by the model.

Support Vector Machine (SVM) was used by Mahjoobi and Mosabbeb (2009) for the prediction of significant wave height. The data was collected from deep water locations in Lake Michigan. They used current and previous six hours wind speed as input variables and significant wave height (H_s) as output parameter. The SVM results were compared with results of multilayer perceptron (MLP), ANN, and radial basis function (RBF) models. They found that SVM could be used successfully for the prediction of H_s. The error statistics of SVM model marginally outperforms ANN with less computational time.

GP has nowadays attracted the attention of researchers in the predictions of hydraulic data (Guven et al., 2009). They used field measurements to develop linear genetic programming (LGP) model for prediction of scour depth around a circular pile due to waves in medium dense silt and sand bed. The LGP model was compared with ANFIS model results. LGP model was observed to be in good agreement with measured data, and quite better than ANFIS and regression based equation of scour depth at circular piles.

Seasonal beach profile evolution was predicted by Hashemi et al. (2010) using ANN at several locations along the Tremadoc Bay. The beach profile data collected in 19 stations for a period of 7 years were studied using ANN. Principal Component Analysis and correlation analysis were used to develop a proper input dataset. The local wave climate, wind data, geometric properties of the beach, and the corresponding beach level changes were fed to a feed forward back propagation

ANN. The field data and the model results were compared. The model resulted predicted the beach surveys with a RMSE of less than 0.0007.

Jian-sheng and Long (2009) proposed a hybrid Particle Swarm Optimization algorithm based on ANN (PSO-BP) to predict the mean monthly rainfall of the Guangxi area. Since, developing the structure of BPNN in meteorological application is difficult this hybrid was used. PSO was used for optimization of number of hidden nodes and connection weights of BP. From their results, it was observed that the hybrid model could effectively improve learning and also help in the generalization of the neural network.

Aydogan et al. (2010) studied to predict vertical current profiles of a given point in a narrow strait, the Strait of Istanbul, using the Feed Forward Back Propagation ANN technique (FFBP-ANN). The model was built on 7039 hours of concurrent measurements of current profiles, meteorological conditions and surface elevations. The model predicted 12 outputs of East and North velocity components at different depths in a given location. Various alternative models with different inputs and neuron numbers were evaluated for attaining the best model by trial and error. Predictions from proposed ANN model were in accordance with the observations with average root mean square error of 0.16 m/s. The same input parameters were then used to build models that predicted current velocities 1–12 h into the future. Results of these predictions showed good overall agreement with observations and that FFBP ANN could be used as a reliable tool for forecasting current profiles in straits.

Balas et al., (2010) applied hybrid model for the preliminary design of rubble mound breakwater and a better agreement between the predicted and measured was obtained as compared to stability equations of Vander Meer's (1988) and ANN.

Chen et al., (2010) proposed the application of Kernel Principal Component Analysis (KPCA) to SVM for feature extraction. They adopted PSO Algorithm to optimize SVM parameters. The novel time series analysis model integrates the advantage of

wavelet, PSO, KPCA and SVM. Compared with other predictors, their model showed greater generality ability and higher accuracy.

Iglesias et al. (2010) developed an AI model to predict the draft required or the draft available if a floating boom would be provided in open waters. The dataset was obtained from laboratory investigations on seven model booms subjected to various wave and current combinations. The obtained dataset was divided into two random sets, one for training and other for testing. The model was further improved to find efficient network architecture using 640 NNs. The model was trained and then generalized them to other cases for validation.

The determination for hyper-parameters including kernel parameters and the regularization is important to the performance of SVM. PSO is a method for finding a solution of stochastic global optimizer based on swarm intelligence. Using the interaction of particles, PSO searches the solution space intelligently and finds out the best one. Pan et al. (2010) built a model to forecast the hydrogen producing reactor temperature using global optimization of PSO and SVM's local accurate searching. They found that the developed model to be feasible and effective. Comparison studies, by the authors, with other existing methods showed that the developed model was highly accurate and effective.

Using ANN and SVM, two nonlinear time-series models were developed by Yoon et al. (2011) for predicting groundwater level (GWL) fluctuations. The models were tested for two coastal aquifer wells in Korea. During training and testing of the models, it was observed that ANN model showed better performance with low RMSE. But, during the prediction stage, SVM performed similar or even better than the ANN model. The input data was superior in SVM model due to its generalization ability. The uncertainty analysis of the two models showed higher uncertainty for ANN model. They concluded that ANN must be used carefully for GWL forecasting in coastal aquifer.

Kim et al., (2011) predicted the stability number of armor blocks of breakwaters using Support Vector Regression. The proposed method proved to be an effective tool for designers of rubble mound breakwaters to support their decision process and to improve design efficiency.

Patil et al., (2011) used ANFIS model for predicting wave transmission coefficient of horizontally interlaced multilayer moored floating pipe breakwater (HIMFPB). It showed that the ANFIS model outperformed ANN model for predicting wave transmission coefficient.

Shi and Dong (2011) used SVM to predict the 28 days strength of cement. The seven input variables used were the content of slag, cement fineness, SO_3 content, 1-day compressive strength and folding strength, 3-day compressive strength and folding strength. The compressive strength predicted by ANN methods were compared with the results of SVM model. The results indicated that SVM has an ability to predict the 28-day strength of cement with an acceptable degree of accuracy (R=0.979, RMSE=0.80, MAE=0.013). They concluded that proposed SVM model was better than the ANN model.

Ahmadi et al. (2011) estimated gas-oil MMP using a model based on a feed-forward ANN optimized by Stochastic Particle Swarm Optimization (SPSO). Initial weights of the neural network were decided using SPSO. The performance of the SPSO-ANN model was compared with the field data and was found to be an effective model.

Tripathy et al. (2011) applied ANN and PSO model for weather prediction based on time series data. The parameters considered were rainfall, maximum and minimum temperature, humidity etc. They tried ANN and PSO tools for the prediction of future weather condition. They developed different training and testing to obtain the accurate result. Their experimental results showed that the proposed model was useful for weather forecasting.

The traditional methods of determination of permeability were core analysis and well test techniques which were costlier and time consuming. Hence, to overcome this problem Gholami et al. (2012) utilized the SVM for predicting the permeability of three gas wells in the southern pars field. The SVM model produced good results with a correlation coefficient of 0.97 for testing dataset. Comparing the results of SVM with the neural network they concluded that SVM approach was faster and more accurate than neural network in prediction of permeability.

Das et al. (2012) implemented Genetic Algorithm (GA) with Support Vector Machine (SVM) in region sampling for selection of local features in handwritten digit recognition. Seven sets of local regions were randomly generated and GA was used to select an optimal group of local regions which produces best recognition performance with SVM. Stimulated Annealing (SA) and Hill Climbing (HC) were also used to find the optimal group of local regions where the maximum accuracies were found to be 97%, 96.7%, 96.7% for GA, SA, HC respectively.

Xi et al. (2012) applied PSO-Neural Network model in prediction of groundwater level in Handan city based on the available old data and their influence factors. Their developed method was better than the grey method. The random characteristics of ground heads in Handan city were analyzed by using PSO-NN. Their results indicated that the method was reliable and reasonable.

For the prediction of reflection coefficient for variety of coastal and harbor structures an ANN model was developed by Zanuttigh et al. (2013). They used 600 datasets for training and testing of the model. The datasets included smooth rock armor units, slopes, berm breakwaters, vertical walls, low crested structures and oblique wave attacks. They used 13 input elements to represent the various aspects of the reflection process. The algorithms and input elements were selected based on the sensitivity analysis. The network model developed was compared with the limited datasets selected from the literature and they found that the model could accurately predict the measured reflection coefficients than the available empirical formulae.

Reliability based risk analysis for rubble mound breakwaters was carried out by Koc and Balas (2013) using fuzzy Monte Carlo simulation approach. They applied fuzzy random variable in the simulation. The results of the study showed that the developed simulation model would prove to be a useful tool in handling both the randomness and fuzziness of the analysis.

ANNs with different topologies were evaluated to predict hydrodynamic coefficients of permeable panelled breakwater (Hagras, 2013). Two neural network models were constructed to predict wave transmission coefficient (Kt) and wave reflection coefficient (Kr). Back propagation algorithm was used to train a multi-layer feed-forward network (Levenberg Marquardt algorithm) and coefficients were evaluated using the Mean Squared Error (MSE). The results of the developed ANN models proved that this technique was reliable. A good match between the measured and predicted values was observed with correlation values varying in the range (0.9508-0.9805) for the training set and (0.9159-0.9877) for the testing set.

From the review of literatures on applications of soft computing in coastal/ocean engineering it is observed that ANNs are commonly used by many researchers to evaluate or predict ocean wave parameters like wave height, wave period, wave direction, tidal levels and its timings, sea levels, water temperature, wind speeds, etc. Damages of coastal structures, stability of breakwaters, storm surges and wave transmission have also been predicted using ANNs. For calibration and verification of the neural network model, many researchers used in-situ data, experimental data, and data generated by numerical or mathematical analysis. It is also found that in most of the applications a three layered feed forward back propagation neural network was used. Some researchers trained the network with conjugate gradient and cascade correlation.

Comparing to the traditional methods, the performance of ANN was improved in terms of computational effort and time needed for training and testing. It is also reported that ANN model can learn with much less data set. Sometimes single

network may not always fit the entire domain of the training sample and in such cases different networks are developed over different sub-domain of the training sample size.

Many researchers have highlighted that hybrid models like ANN with fuzzy, genetic algorithm can be adopted when the performance of individual ANN model is poor in mapping input-output relation. Apart from ANNs, other new techniques like Support vector machines, Particle Swarm Optimization or combinations of these can be used to solve problems in coastal engineering.

2.4 PROBLEM FORMULATION

The literature review work carried on theoretical and experimental analysis on breakwater revealed that the researchers have carried out number of studies by considering simple form of berm breakwater and some common assumptions in hydrodynamics to derive mathematical model for stability and damage. But, these mathematical models showed very poor agreement with experimental or in-situ data. Waves in nature are random and hence, very complicated to perform physical model study on breakwaters. Also, the physical model studies are expensive and time consuming. Further, there is no simple mathematical model to predict the damage level considering all the boundary conditions due to the complex nature of the problem which includes wave structure interaction, angle of wave attack, movement of the armor etc.

From the literature review carried it was observed that soft computing techniques have been successfully used in coastal engineering solve complex problems. However, it was also observed that soft computing techniques for prediction of damage level of a non-reshaped berm breakwater have hardly been used.

In view of the above aspects, a detailed study was taken up on developing soft computing techniques for the prediction of damage level of non-reshaped berm breakwater, thereby providing a new approach to solve the problem of damage level

prediction, which is highly complex and has huge non-linearity associated with its hydrodynamic performance.

2.5 OBJECTIVES OF THE PRESENT INVESTIGATION

The objectives of the present investigation involve the development of soft computing models to predict the damage of non-reshaped berm breakwater and are listed as follows:

- To investigate the ability of soft computing approaches like Artificial Neural Networks (ANN) and support vector machines (SVM) to effectively address various hard-to-solve design tasks and issues associated with the berm breakwater.
- To identify the most significant parameters as input to soft computing models by using principal component analysis.
- To develop soft computing models for prediction of damage level of non-reshaped berm breakwater.
- To integrate and hybridize the fuzzy logic, ANNs, SVMs and PSOs to improve the damage level predictions of non-reshaped berm breakwater.
- To analyze and recommend the most reliable soft computing model in predicting damage level of non-reshaped berm breakwater.

2.6 SUMMARY

Detailed literature review on the experimental and theoretical works on breakwaters has been discussed in this chapter**.** It is observed that these theoretical studies fail to give a mathematical model to predict damage level of breakwater by considering all boundary conditions. This is because of complexity and non-linearity associated with wave-structure interaction. Also it is brought to the notice that physical model studies are time consuming. To overcome this problem, soft computing techniques is successfully used to solve the issue of complexity and non-linearity. A detailed literature review on applications of soft computing techniques in coastal/ocean engineering has been discussed including the latest available literature. However, it is observed that there are hardly any applications of soft computing techniques on

prediction of damage level of breakwater. Hence, there is a great scope for developing soft computing models in prediction damage level of breakwater. The objectives of the present investigation are also discussed in this chapter.

CHAPTER 3

EXPERIMENTAL DATA AND METHODOLOGY

3.1 GENERAL

A physical model study on statically non-reshaped berm breakwater was carried out (Rao et al., 2004 and 2008; Balakrishna Rao, 2009) in a two-dimensional wave flume. The model was designed to suit the wave parameters off- Mangalore coast. The data obtained by them has been used in the present research work.

In the present work, experimental data obtained from the physical model study was used for developing soft computing models to predict the damage level of non-reshaped berm breakwater. Experimental data were collected and organized in a systematic data base. These data were divided into two sets, about 80% of the data for training and remaining data for testing models.

In this chapter details of the experiments are explained along with the methodology adopted for the present study. Also, details about Artificial Neural Network (ANN), Support Vector Machines (SVM), Adaptive Neuro-Fuzzy Inference System (ANFIS) and Particle Swarm Optimization (PSO) are described in this chapter.

3.2 EXPERIMENTAL SETUP

3.2.1 Wave flume

The physical model was tested for regular waves in a two dimensional wave flume facility available in the Marine Structures Laboratory of Department of Applied Mechanics and Hydraulics, National Institute of Technology Karnataka, Surathkal, India. The wave filter consists of a series of vertical asbestos cement sheets spaced at about 0.1m centre to centre parallel to length of the flume. The wave flume is 50m long, 0.71m wide and 1.1m deep. It has a 41.5m long smooth concrete bed. About

15m length of the flume is provided with glass panels on one side. It has a 6.3m long, 1.5m wide and 1.4m deep chamber at one end where the bottom hinged flap generates the waves. The flap is controlled by an induction motor of 11kW power at 1450rpm. This motor is regulated by an inverter drive (0 – 50Htz) rotating in a speed range of 0–155rpm. Fig. 3.1 gives a schematic diagram of experimental setup. Plate 3.1 and Plate 3.2 shows different view of wave flume. By changing the frequency through inverter one can generate the desired wave period. A fly-wheel and bar-chain link the motor with flap. By changing the eccentricity of bar chain on the fly-wheel one can vary the wave height for a particular wave period (Balakrishna Rao, 2009).

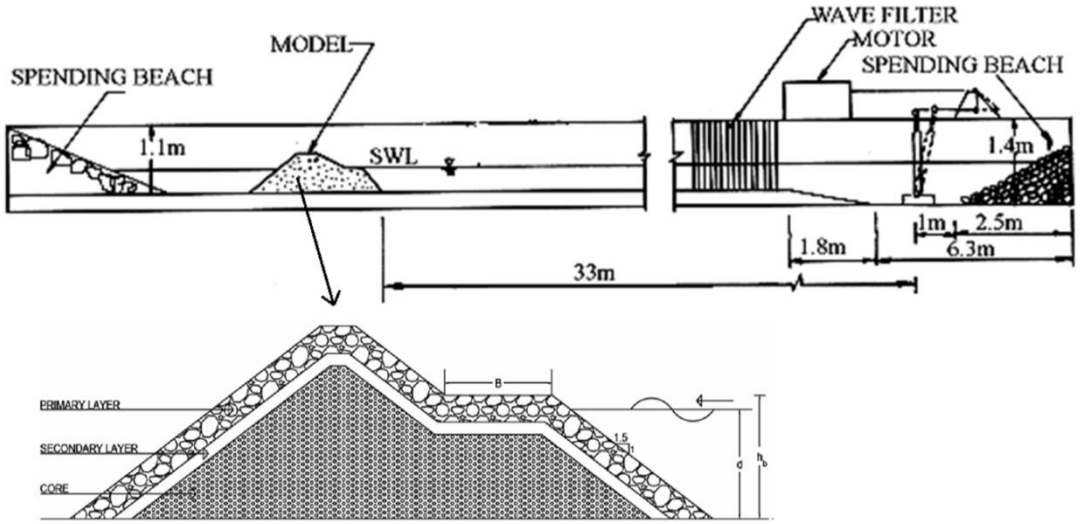


Fig. 3.1: Wave Flume with Berm Breakwater

Plate 3.1 View of wave flume with model and wave probe

Plate 3.2 View of wave flume from model end

The data was collected using wave probes and data acquisition system, and the damage level was measured using profiler system. Plate 3.3 and 3.4 shows the wave probe system and the profiler system used in the experimental study.

Plate 3.3 View of wave flume with wave probes

Plate 3.4 View of wave flume with profiler

3.2.2 Experimental data collected

The data is collected from the experiments conducted by Balakrishna Rao (2009). The details of physical study conducted and the model details are explained in following paragraphs.

Totally four set of experiments (set no. I, II, III, and IV), were conducted (Rao et al. 2004, 2008). First set of experiments were conducted on conventional breakwater model with armor stone weight W_{50} = 74 g, which was calculated using Hudson equation for a design wave height of 0.1 m (K_D = 3.5, cotα = 2, ρ_a = 2.74, ρ_w = 1.0, two layer and random placement).

In the second set, statically stable non-reshaped berm breakwater models were tested. The weight of median armor stones (W_{50}) in these models was kept at 52 g which is about 30% less than 74 g, the weight used in conventional breakwater. In this set of experiments the influence of berm width on the stability of the breakwater, wave run-up and run-down was studied.

In the third set of experiments the influence of tidal effect on the stability of statically stable non-reshaped berm breakwater model was investigated. The depth of water in front of the breakwater model was varied and the stability was tested against different wave parameters. The weight of armor stones in these models was W_{50} = 58.6 g which is about 20% less than 74 g, the weight used in conventional breakwater.

In the fourth set of experiments the influence of location of the berm on the stability of the statically stable non-reshaped berm breakwater model was studied. The location of the horizontal surface of berm was varied and the stability was tested against different wave parameters. The weight of armor stones in these models were W_{50} = 52 g which is about 30% less than 74 g, the weight used in conventional breakwater.

The various sea state and structural parameters considered by Balakrishna Rao (2009) for the experimental study is given in Table 3.1. He also performed non-dimensional analysis to arrive at some of the dimensionless parameters which is used in the present study. The dimensionless parameters obtained were deepwater wave steepness (H_o/L_o), surf similarity parameter (ξ), relative berm position (h_B/d), armor stone weight (W_{50}/W_{max50}), relative berm width (B/L_o) and relative berm location (h_B/L_o). The variation of non-dimensional parameters obtained above is shown in Fig. 3.2.

Table 3.1 Range of experimental variables

Variable	Range
Wave height, H (m)	0.10, 0.12, 0.14, 0.16
Wave period, T (s)	1.6, 2.0, 2.6
Water depth above the bed level, d (m)	0.25, 0.30, 0.35, 0.40
Water depth above or below the berm, d_B (m)	+0.08,+0.03, -0.02,-0.07
Armor stone weight, W_{50} (g)	52 to 74
Crest height above the seabed (m)	0.70
Berm width, B (m)	0.60
Berm position above seabed, h_B (m)	0.32

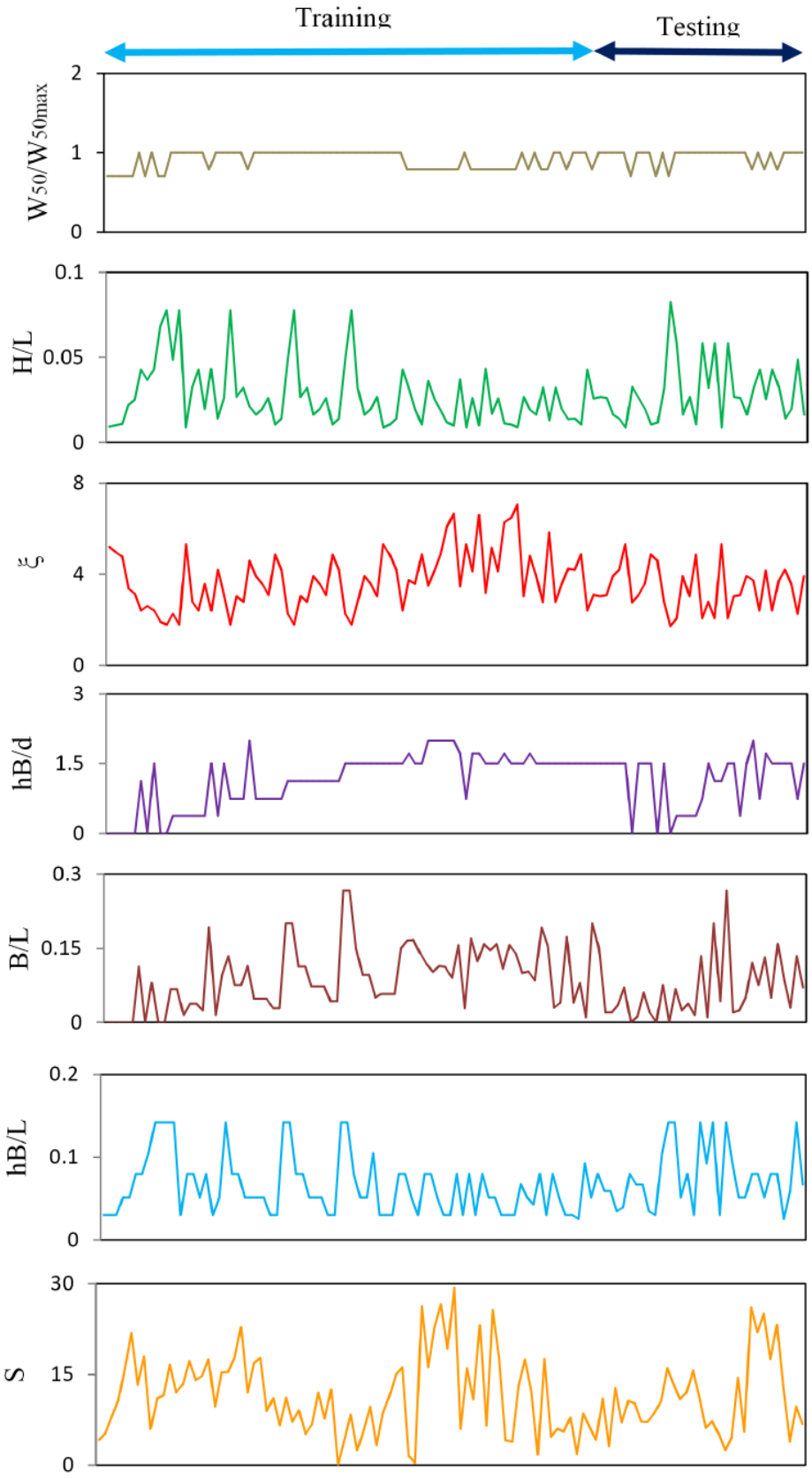


Fig. 3.2 Experimental data on H/L_0, ξ, h_B/d, W_{50}/W_{50max}, B/L_0, h_B/L_0 and S

3.4 ARTIFICIAL NEURAL NETWORK

An artificial neural network is simplified model of the biological neuron system (Fig. 3.3). It is a massively parallel distributed processing system made up of highly interconnected neural computing elements. They have the ability to learn from examples and acquire knowledge and make it use when it required. In other words, neural network function in a way similar to the human brain.

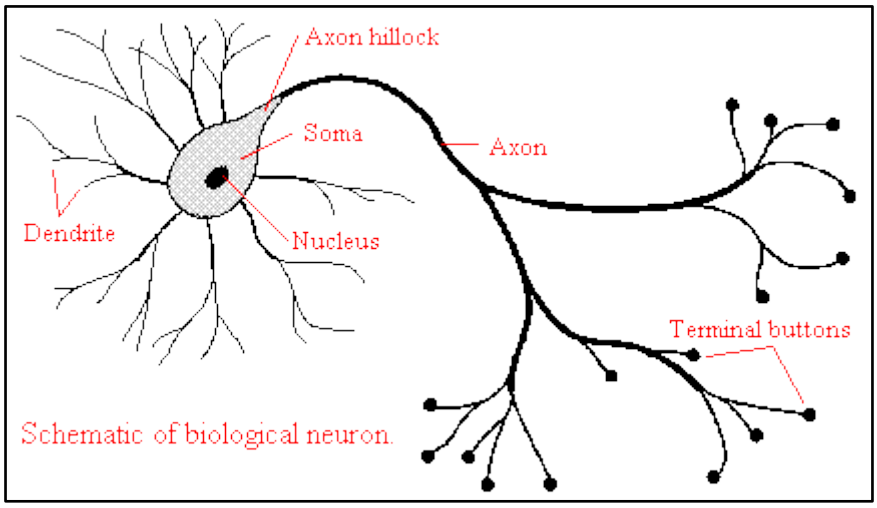


Fig. 3.3 A biological nerve cell

A biological neuron has three types of components namely dendrites, soma and axon. The dendrites receive signals from other neurons. The soma sums the incoming signals. When sufficient input is received, the cell fires. The output area of the neuron is a long fiber called axon. The impulse signal triggered by the cell is transmitted over the axon to other cells. The connecting point between a neuron's axon and another neuron's dendrite is called a terminal buttons or synapse. The impulse signals are then transmitted across a synaptic gap by means of a chemical process.

The artificial neuron mimics the characteristics of the biological neuron. The artificial neuron basically consist inputs, each inputs represents the output of another neuron (Fig 3.4). The amount of information about the input that is required to solve a problem is stored in the form of weights. Each input is multiplied with an associated weight before it reaches to the summing node. In addition, the artificial neuron has a bias term, a threshold value that has to be reached or extended for the neuron to

produce a signal, a nonlinear function (F) that acts on the produced signal net and output (Y) after the nonlinearity function.

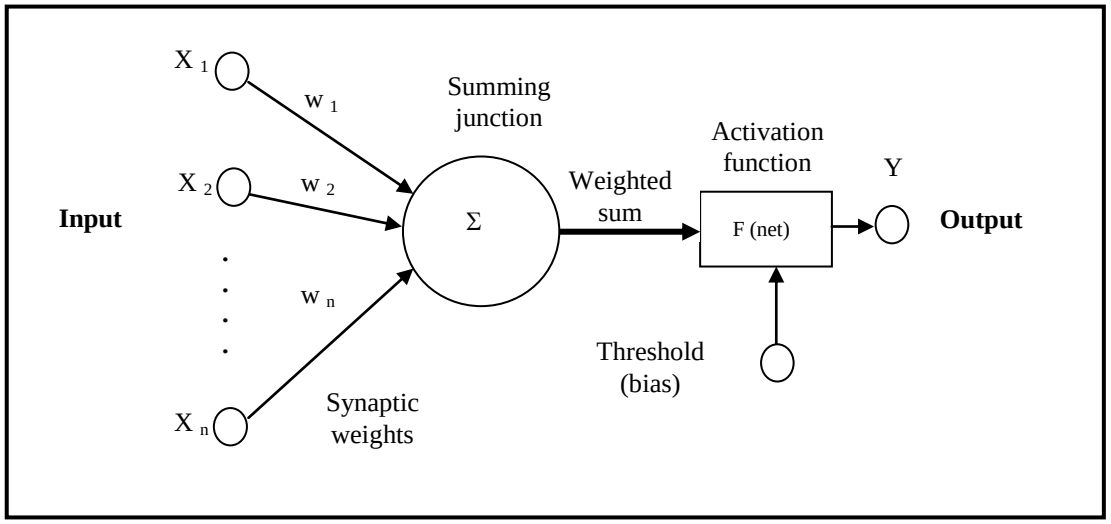


Fig. 3.4 Basic Neuron Model

Many research activities have been carried out based on this method in different fields (Mase, et al., 1995; Yagci, et al., 2005; Mandal et al, 2005; Kim and Park, 2005; Hashemi et al., 2010; Aydogan, et al., 2010; Iglesias, et al., 2010). The most significant features of neural networks are the extreme flexibility due to learning ability and the capability of nonlinear function approximations. This fact leads us to expect neural networks to be an excellent tool for solving the motion characteristics of the breakwater while overcoming complexity and non-linearity associated with wave-structure interaction of berm breakwater.

3.3.1 Architecture of an ANN

The neurons are assumed to be arranged in layers, and the neurons in the same layer behave in the same manner. All the neurons in a layer usually have the same activation function. Within each layer, the neurons are either fully interconnected or not connected at all. The neurons in one layer can be connected to neuron in other layer. The arrangement of neurons into layers and the connection pattern within and between layers is known as network architecture.

Input layer: The neurons in this layer receive the external input signals and perform no computation, but simply transfer the input signals to the neurons in another layer.

Output layer: The neuron in this layer receive signals from neurons either in the input layer or in the hidden layer

Hidden layer: The layer of neurons that are connected in-between the input layer and the output layer is known as hidden layer.

3.3.2 Feed Forward Back-Propagation Neural Network

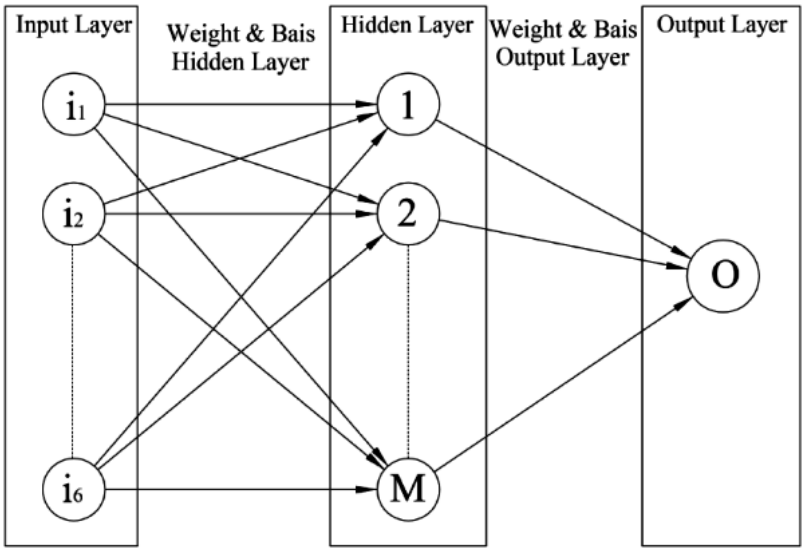


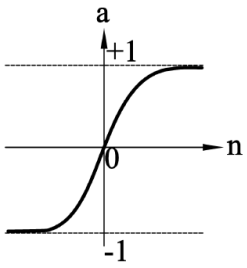
Fig.3.5 Architecture of Neural Network

Fig.3.5 shows the architecture of neural network. The Feed Forward Network (FFN) commonly used for supervised learning which consists of three layers, namely I-number of nodes in input layer, M-number of nodes in hidden layer and O- number of nodes in output layer. These three layers are highly interconnected by nodes and work together to solve specific problems. Once the data is received in input layer the processed values are sent to the hidden layer. The hidden layer and output layer process all incoming signals by applying some weights to them. The Feed Forward Network (FFN) uses back propagation for training network where the error at the output layer is moved back to the input-hidden layers for updating weights and decrease errors to yield best results. The main aim of the FFN process is to reduce the overall error (E) between the observed and predicted values by adjusting the weights. And these weights are combined and processed through an activation function and released to the output layer.

The activation functions used in hidden layer and output layer are Tansig and Purelin.

Tansig:

It is a hyperbolic tangent sigmoid function whose output y ranges between -1 and +1 as input x varies from -∞ to +∞ as shown in Fig 3.6. This function allows negative values of output also.

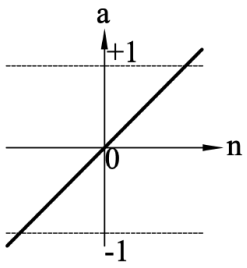


a=tansig(n)

Fig. 3.6 Hyperbolic tangent sigmoid Activation Function

Purelin:

It is purely a linear function ($y = mx + c$) as shown in Fig 3.7. It does not limit the amplitude of the output and the output varies linearly with the amplitude of the input.



a=purelin(n)

Fig. 3.7 Purelin Activation Function

The network is trained by using different algorithms namely, Scaled Conjugate Gradient (SCG), Gradient descent with adaptive learning rate backpropagation (GDA) and Levenberg-Marquardt (LM).

3.3.3 Scaled Conjugate Gradient

Conjugate gradient is the most popular iterative method for solving large systems of linear equations. Conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration. SCG is a second order conjugate gradient algorithm that helps minimize goal function of several

variables. These theoretical foundations was proved by Moller (1993) which remains first order techniques in first derivatives like standard backbropagation and find the better way to a local minima in second order techniques in second derivatives. SCG use a step size scaling mechanism avoids a time consuming line-search per learning iteration, which makes the algorithm faster than other second order algorithms recently proposed. Based on the Moller (1993), SCG method shows super linear convergence on most problems.

Conjugate gradient iteration is described as

$$X_k = X_{k-1} + \alpha_k d_{k-1} \tag{3.1}$$

K is iteration index, αk is the length of the step performed at iteration, dk is the search direction, rk is residual vector and βk is improvement.

$$\alpha_k = \left(r_{k-1}^T r_{k-1}\right) / \left(d_{k-1}^T A d_{k-1}\right) \tag{3.2}$$

$$d_k = r_k + \beta_k d_{k-1} \tag{3.3}$$

$$r_k = r_{k-1} - \alpha_k A d_{k-1} \tag{3.4}$$

$$\beta_k = \left(r_k^T r_k\right) / \left(r_k^T - 1 r_{k-1}\right) \tag{3.5}$$

Above Eqs. 3.2-3.5 shows the relative component of approximation solution for conjugate gradient**.**

3.3.4 Gradient Descent with Adaptive Learning Rate Backpropagation

The algorithm was used to updates weight and bias values according to gradient descent with adaptive learning rate. An adaptive learning rate tries to keep the learning step size as large as possible while keeping learning stable. The learning rate is made responsive to the complexity of the local error surface.

An adaptive learning rate requires some changes in the training procedure used by traingd. First, the initial network output and error are calculated. At each epoch new

weights and biases are calculated using the current learning rate. New outputs and errors are then calculated.

As with momentum, if the new error exceeds the old error by more than a predefined ratio, the new weights and biases are removed. In addition, the learning rate is decreased. Otherwise, the new weights and biases are kept. If the new error is less than the old error, the learning rate is increased.

This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. Thus, a near-optimal learning rate is obtained for the local terrain. When a larger learning rate could result in stable learning, the learning rate is increased. When the learning rate is too high to guarantee a decrease in error, it is decreased until stable learning resumes.

3.3.5 Levenberg-Marquardt (LM) Algorithm

The LM algorithm is a second-order method (Hagan and Menhaj, 1994; Masters, 1995). Rather than finding the error minimum directly, it aims to locate the zero of the error gradient. The zero α of a univariate function *f* may be found using the Newton-Raphson method according to the iterative formula of Eq 3.6

$$\alpha_{n+1} = \alpha_n - \frac{f(\alpha_n)}{f(\alpha_n)} \tag{3.6}$$

When extended to a multivariate function, α becomes a vector and the derivative of the function is now a vector derivative, as in Eq 3.7.

$$\alpha_{n+1} = \alpha_n - \frac{f(\alpha_n)}{\nabla(\alpha_n)} \tag{3.7}$$

In the case of neural network optimization, we wish to find the zero of the error gradient g with respect to the network weights. Since g is a vector quantity and is itself a derivative, we have to work with the Hessian matrix H (Eq 3.8).

$$w_{n+1} = w_n - [g(w_n) / H(w_n)] \tag{3.8}$$

Each element in the Hessian contains second derivatives of the error function, summed over all training patterns. However, the error measure *E* is related to the outputs and target outputs. The elements within the Hessian therefore contain values like that in Eq 3.9, summed across all training patterns.

$$\frac{\partial^2 E}{\partial w_i \partial w_j} = 2\left(\frac{\partial_y}{\partial w_i}\frac{\partial_y}{\partial_{wj}} + (y-t)\frac{\partial^2 y}{\partial w_i \partial w_j}\right) \tag{3.9}$$

One can calculate local values of the first derivatives. These are the ∂ values used in the gradient descent method. The second derivatives in the above equation are disregarded when estimating the Hessian. This is a reasonable estimate since the error ($y-t$) is expected to be small. Further, we expect the values of ($y - t$) to have an approximately Gaussian distribution with mean zero. When summed over a large number of training patterns the second terms are therefore likely to cancel out to a large extent. Having obtained an approximation of the Hessian, the Newton-Raphson method may be used to find the nearest zero of the error gradient. Two problems may arise. Firstly, the local Hessian estimation may not be an adequate representation of the underlying function. Secondly, the second-order algorithm by itself may approach a maximum or saddle point on the error surface, rather than a minimum. In order to avoid these problems, the LM method includes an additional gradient descent term. The weight adjustment vector is then given by Eq 3.10.

$$\Delta \mathrm{w} = (\mathrm{H} + \lambda\, diag\,(\mathrm{H}))^{-1}\,\mathrm{g} \tag{3.10}$$

The parameter λ adjusts the relative weighting given to Newton's method and to gradient descent. If the error falls after applying the weight adjustment, λ is decreased. If, on the other hand, the error increases, the weight changes are reversed λ is increased and the weight changes are re-calculated.

In the present study the FFN (I-M-O) with seven inputs parameters (I=7), namely $i_1= W_{50}/W_{max50}$, $i_2=H_o/L_o$, $i_3= \xi$, $i_4=B/d$, $i_5=B/ L_o$, $i_6= h_B/L_o$,$i_7=\cot\alpha$, M is the number of

hidden layer nodes, here M=5 and O is one output layer node, here O=damage level (S).The activation function used in the hidden layer is hyperbolic tangent sigmoid transfer function (tansig) and linear transfer function (purelin) in the output layer. This model is developed using different algorithms namely SCG, GDA and LMA. From PCA study it is observed that cotα is the least influential parameter on damage level. Based on the PCA study least influential parameter is discarded and ANN2 model (I_6-M_5-O_1) is developed with remaining input parameters.

3.3.6 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It was adopted to find out the input variables with percentage of influences on damage level. The variables, which have less percentage of influences, were separated out. The network analysis was carried out for the reduced number of variables.

The number of components extracted (created) in a principal component analysis is equal to the number of observed variables being analyzed. However, in most analyses only the first few components account for meaningful amounts of variance so only these first few components are interpreted and used in a subsequent analyses such as a multiple regression. The first principal component accounts for the most possible variance in the data. The second component accounts for the most variance not accounted for by the first component, and so on until all variables are accounted for. The first few components account for most of the total variation in the data, and can be used for subsequent analysis.

The first principal component extracted in a principal component analysis accounts for a maximal amount of total variance in the observed variables. Under typical conditions, this means that the first component is correlated with at least some of the observed variables. In fact, it is often correlated with many of the variables.

The second principal component extracted has two important characteristics.

- The second component accounts for a maximal amount of variance in the data not accounted for by the first component. Under typical conditions, this means that the second component is correlated with some of the observed variables that did not display strong correlations with the first component.
- The second characteristic of the second component is that it is uncorrelated with the first component. If you compute the correlation between component 1 and component 2, that correlation is zero.

The remaining components extracted in the analysis display these same two characteristics each component accounts for a maximal amount of variance in the observed variables that was not accounted for by the preceding components and is uncorrelated with all of the preceding components. A principal component analysis proceeds in this manner with each new component accounting for progressively smaller amounts of variance. This is why only the first few components are retained and interpreted. When the analysis is complete, the resulting components display varying degrees of correlation with the observed variables, but are completely uncorrelated with one another.

3.5 SUPPORT VECTOR MACHINES

The foundation of SVM has been developed by Vapnik (1995) and is gaining popularity due to many attractive features and promising empirical performance. The formulation represents the Structural Risk Minimization (SRM) principle (Gunn, 1997) which has been shown to be superior to traditional Empirical Risk Minimization (ERM) principle as adopted by conventional neural networks. SRM principle minimizes an upper bound on the expected risk, as opposed to ERM principle that minimizes the error on the training data. This difference outfits SVM with a greater ability to generalize, which is the goal in statistical learning. SVMs were developed to solve the classification problem, but recently they have been extended to the domain of regression problems (Vapnik et al., 1996).

Mathematics behind SVM algorithm for regression

Consider a training data set $g = \{(x_1, y_1), (x_2, y_2), ...(x_p, y_p)\}$, such that $x_i \in v^N$ is a vector of input variables and $x_i \in v$ is the corresponding scalar output (target) value. Here, the modeling objective is to find a regression function, $y = f(x)$, such that it accurately predicts the outputs $\{y\}$ corresponding to a new set of input-output examples, $\{(x, y)\}$, which are drawn from the same underlying joint probability distribution as the training set. To fulfill the stated goal, support vector regression (SVR) considers the following linear estimation function Eq.3.11.

$$f(x) = (w \cdot x) + b \tag{3.11}$$

Where, w denotes the weight vector; b refers to a constant known as "bias"; $f(x)$ denotes a function termed feature, and $(w \cdot x)$ represents the dot product in the feature space, l , such that $\phi : x \to l, w \in l.$ The basic concept of support vector regression is to map nonlinearly the original data x into a higher dimensional feature space and solve a linear regression problem in this feature space.

The regression problem is equivalent to minimize the following regularized risk function:

$$R(f) = \frac{1}{n}\sum_i^n L(f(x_i) - y_i) + \frac{1}{2}\|w\|^2 \tag{3.12}$$

Where,

$$L(f(x) - y) = \begin{Bmatrix} \|f(x) - y\| - \varepsilon, \; for \, |f(x) - y| \, x \geq \varepsilon \\ 0, otherwise \end{Bmatrix} \tag{3.13}$$

Eq. 3.13 is also called ε-insensitive loss function. This function defines a ε-tube. If the predicted value is within the ε-tube, the loss is zero or else the loss is equal to the magnitude of the difference between the radius ε of the tube and the predicted value. The radius of the tube located around the regression function (Fig. 3.8) is represented by a precision parameter ε and the "ε-intensive zone" is the region enclosed by the tube. The SVM algorithm attempts to position the tube around the data as shown in Fig.3.8. By substituting the e-insensitive loss function into Eq. (3.12), the optimization object becomes:

$$\textit{Minimize} \qquad \frac{1}{2}\|w\|^2 C + \sum_i^n \left(\xi_i + \xi_i^*\right) \tag{3.14}$$

With the constraints,

$$\textit{Subjected} \qquad \left\{\begin{array}{l} y_i - (w \cdot x) - b \le \varepsilon + \xi_i \\ (w \cdot x) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{array}\right\} \tag{3.15}$$

Where, the constant $C > 0$ stands for the penalty degree of the sample with error exceeding epsilon.

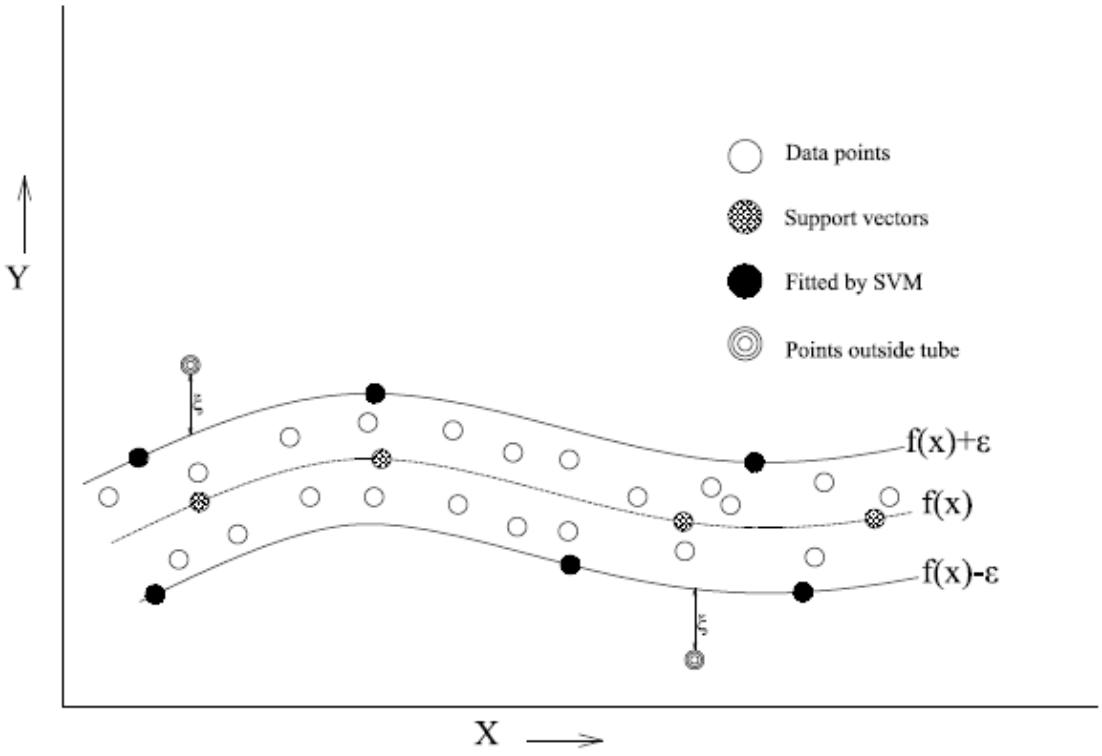


Fig. 3.8 A schematic diagram of support vector regression using e-insensitive loss function

The distance from actual values to the corresponding boundary values of ε-tube is represented by the two positive slack variables.

The SVM fits $f(x)$ to the data in a manner such that:

(i) Minimizing the slack variables i.e., ξ_i, ξ_i^* the training error is minimized and,

(ii) To increase the flatness of $f(x)$ or to penalize over complexity of the fitting function $\|w\|^2$ is minimized.

A dual problem can then be derived by using the optimization method to maximize the function,

Maximize:

$$-\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_j-\alpha_i^*)(\alpha_i-\alpha_j^*)(x_i,x_j)-\varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{n}y_i(\alpha_i-\alpha_i^*) \qquad (3.16)$$

$$\text{Subject to} \quad \sum_{i=1}^{n}(\alpha_i+\alpha_i^*)=0 \quad \text{and} \quad 0\le\alpha_i,\alpha_i^*\le C \qquad (3.17)$$

Where, α_i, α_i^* are Lagrange multipliers. Owing to the specific character of the above-described quadratic programming problem, support vectors (SVs) are the non-zero coefficients, $(\alpha_i-\alpha_i^*)$ corresponding to input vectors x_i. The SVs can be thought of as the most informative data points that compress the information content of the training set. The coefficients α and α^* have an intuitive interpretation as forces pushing and pulling the regression estimate $f(x_i)$ towards the measurements y_i.

The SVM for function fitting obtained by using the above mentioned maximization function is then given by,

$$f(x)=\sum_{i-1}^{n}(\alpha_i-\alpha_i^*)(x_i\cdot x)+b \qquad (3.18)$$

As for the nonlinear cases, the solution can be found by mapping the original problems to the linear ones in a characteristic space of high dimension, in which dot product manipulation can be substituted by a kernel function, i.e. $K(x_i,y_i)=\phi(x_i)\phi(y_i)$. In this work, the SVM is used with different kernel function. Substituting $K(x_i,y_i)=\phi(x_i)\phi(y_i)$ in Eq.3.16 allows us to reformulate the SVM algorithm in a nonlinear paradigm. Finally, we have,

$$f(x)=\sum_{i-1}^{n}(\alpha_i-\alpha_i^*)K(x_i\cdot x)+b \qquad (3.19)$$

In the present work, kernel functions, such as polynomial, radial basis function (rbf), exponential radial basis function (erbf), spline and b-spline are used for non-linear SVM models as shown in Table 3.2. Good setting of kernel parameters (d,γ) and

SVM parameters (C, ε) of SVM model are important in accuracy predicting, where, d represents the degree of polynomial and b-spline kernel functions and γ is the width of rbf and erbf kernel functions. Parameter C determines the trade-off between the model complexity (flatness) and the degree to which deviations are larger than ε tube (Vapnik, 1995 and Smola and Schölkopf, 1998). Parameter ε controls the width of the ε-insensitive zone, which is used to fit the training data. The number of support vectors (nsv) used to construct regression function depends on ε . For the higher ε, the fewer support vectors are selected and hence results in data compression (Kecman, 2001). The performance of SVM regression depends on the good setting of SVM and kernel parameters. In the present study, quadratic loss function is used. The main idea of using this loss function is to ignore the errors, which are situated within the certain distance of the true value.

Table 3.2 Kernel Functions

Kernels	Functions
polynomial	$K(x_i, x_j) = ((x_i, x_j)+1)^d$
rbf	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right)$
erbf	$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\gamma^2}\right)$
spline	$K(x_i, x_j) = 1 + (x_i, x_j) + \frac{1}{2}(x_i, x_j)\min(x_i, x_j) - \frac{1}{6}\min(x_i, x_j)^3$
b-spline	$K(x_i, x_j) = B_{2d+1}(x_i - x_j)$

3.6 PARTICLE SWARM OPTIMIZATION

PSO is a population-based stochastic optimization technique motivated by social behavior, such as bird flocking and fish schooling. PSO was first proposed by Kennedy and Eberhart (1995).

In PSO, each single candidate solution is an “individual bird of the flock", that is a particle in the search space. Each particle makes use of its individual memory and knowledge gained by the swarm as a whole to find the best solution. All the particles have fitness values, which are evaluated by fitness function to be optimized and have velocities which direct the movement of the particles. The best position of each particle is chosen from its own and neighboring particle experience in the process of movement of the particles. The particles move through the problem space by following a current of optimum particles. The initial swarm is generally created in such a way that the population of the particles is distributed randomly over the search space. Every iteration each particle is updated by following two "best" values called *pbest* and *gbest*. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) achieved so far. This fitness value is stored, and called *pbest*. When a particle takes the whole population as its topological neighbor, the best value is a global “best” value and is called *gbest*.

The PSO algorithm is defined by the direction and movement of each particle through the search space, by updating its
velocity:

$$Vel_{j+1}^{i} = W_j Vel_j^i + C_1 rand_1 \left(pbest_j - pos_j^i \right) + C_2 rand_2 \left(gbest_j - pos_j^i \right) \qquad (3.20)$$

and position:

$$pos_{j+1}^{i} = pos_{j+1}^{i} + Vel_{j+1}^{i} \qquad (3.21)$$

where pos_j^i is the current position of the particle i with subscript j representing iteration count, Vel_j^i the search velocity of the i^{th} particle, C_1 and C_2 the cognitive and social scaling parameters, $rand_1$ and $rand_2$ are the random numbers on the interval [0, 1] applied to the i^{th} particle, W_j is the particle inertia, $pbest_j$ is the best position found by the i^{th} particle (personal best), and $gbest_j$ is the global best position found among all the particles in the swarm. The number of companions will affect the

convergence speed of the algorithm. The cognitive component represents learning achieved from an individual particle's search come across. In contrast, the social scaling parameter represents the cooperation among companions learning from search experience.

$$W_j = w_{Max} - [(w_{Max} - w_{\min}) \times iter] / \max_{Iter} \tag{3.22}$$

where w_{Max} = initial weight, $w_{\min}$ = final weight,

$\max_{Iter}$ = maximum iteration number, $iter$ = current iteration number.

The particle inertia controls the balance of global and local search abilities, where a larger inertia weight (w) helps for global search while a small inertia weight helps for a local search. By decreasing the inertia weight linearly from larger value to a smaller value through the course of the PSO run gives the best PSO performance compared with fixed inertia weight settings. Particle i flutters towards a new position using Eq. (3.20) and Eq. (3.21), which allow all particles in the swarm to update their $pbest_j$ and $gbest_j$.

The algorithm steps are as follows:

Step1: Initialize PSO parameters particle positions and velocities;

Step2: Objective function values are calculated using the initial particle positions and velocities;

Step3: Update the optimum particle positions and global optimum particle position using fitness function;

Step4: Update the position of each particle and the velocity vector by using Eq. (3.20) and Eq. (3.21);

Step5: Repeat step 2 to 4 until the stopping criteria are met.

3.6.1 Proposed PSO-SVM model

To avoid over-fitting or under-fitting of the SVM model due to the improper selection of SVM and kernel parameters PSO is used to select suitable parameters of SVM. PSO is an optimization method, which not only has strong global search capability, but also is very easy to implement. In the present study, MATLAB support vector

machine toolbox (Math Works 2011) is interfaced with particle swarm optimization to optimize the SVMs and kernel parameters simultaneously for better generalization of the proposed PSO-SVM model.

PSO-SVM model was developed by using different kernel functions as shown in Table 3.2. To study the performance of each kernel, PSO-SVM models were developed to predict damage level of non-reshaped berm breakwater. Experimental data set was divided into two groups one for training and other for testing. Fig.3.9 illustrates the proposed PSO-SVM model. Initially, training input, training target, kernel function, and kernel and SVM parameters are given as input to the system. PSO generates initial particle position and velocities that would be used to find optimum factors of kernel and SVMs parameters. In the second step, typical SVM process performs using initial values and evaluates the objective function using initial particle position. Here, the fitness value for each particle position and velocities are called as pbest. When a particle takes the whole population as its topological neighbor, the best value is a global "best" value and is called gbest. The main objective of the present study is to find optimal parameters which give good results.

In third step the fitness value are calculated and is compared with pbest and gbest, if it satisfies the terminal condition in PSO, then these parameters are selected as optimal, otherwise, the Eq. (3.20) and Eq. (3.21) are used to update the position and velocities. After new position and velocities are updated, the training process is performed again. Step two and three are iterated again and again until the stopping conditions are satisfied. Once the stopping condition is satisfied, the population search ends, and the pbest and gbest that show the best performance in the last step3 are selected as the final result. Finally, optimized SVMs and kernel parameters obtained by PSO are tested using test data to check the performance of model.

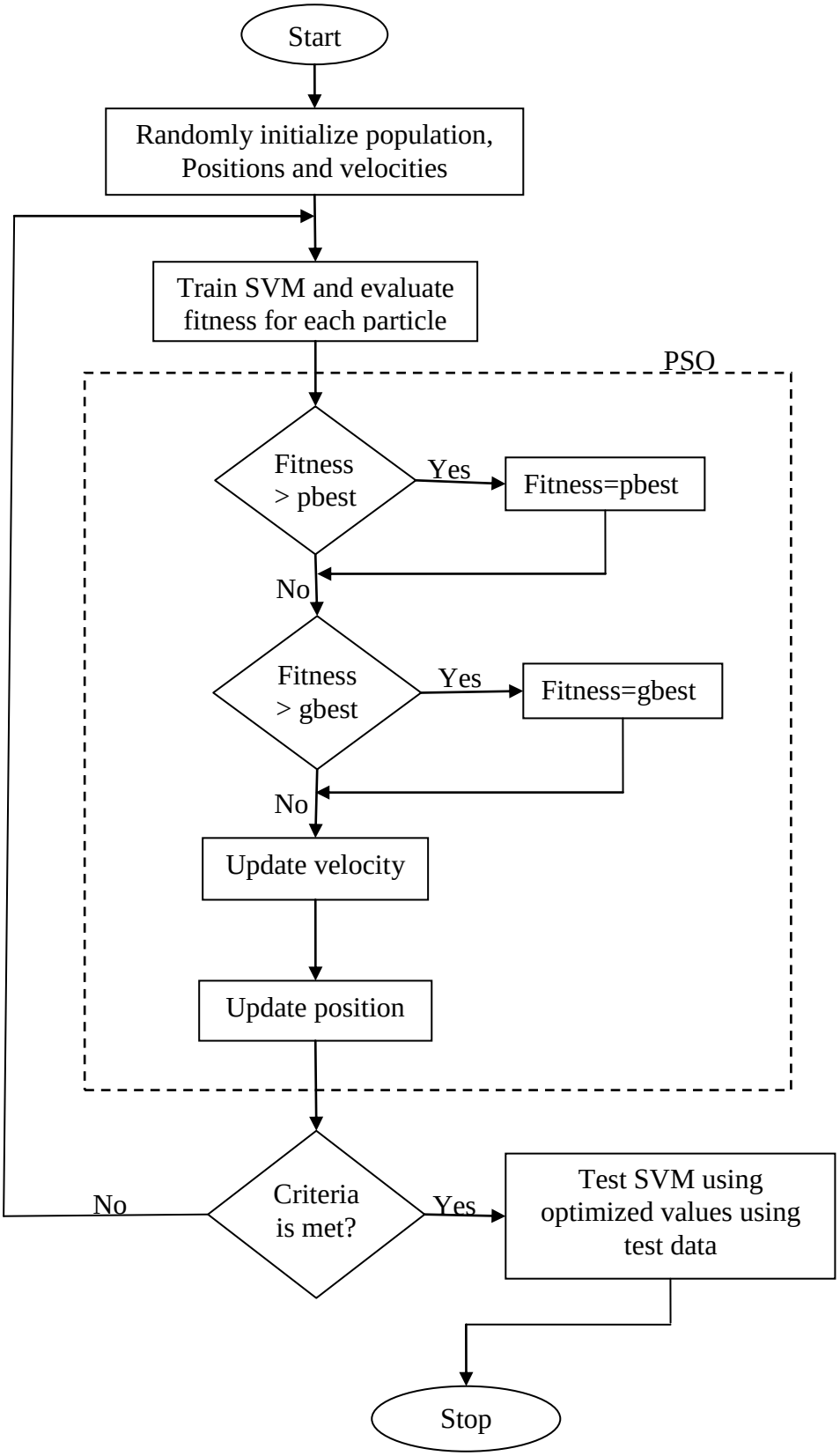


Fig. 3.9 Flow chart of PSO-SVM for solving optimization problems

3.7 ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM

Fuzzy logic has a great advantage of representing knowledge very precisely giving good explanation and reasoning in a understandable manner. ANN is fault tolerant and it has a great capability of learning on its own. But, the representation of knowledge stored is in the form of connection between the neurons which is difficult to understand. Hence if both the techniques are combined, a huge advantage of self learning and an understandable knowledge representation is possible. Further, ANN requires crisp data and preparation of such data is tedious and time consuming. This can be avoided by developing an interface that directly inputs the environmental fuzzy data which can be done by using fuzzy logic. This combination is the neuro-fuzzy approach popularly called as Adaptive Neuro Fuzzy Inference System (ANFIS) has been developed.

ANFIS was first introduced by Jang (1993). It is a combination of least-squares and back-propagation gradient decent methods used for training Takagi-Sugeno type fuzzy inference system which is used for an effective search for the optimal parameters. It can provide a starting point for constructing a set of fuzzy 'if-then' rules with appropriate membership functions to generate the fixed input-output pairs. ANFIS is a simple structure with effective learning algorithm and high speed (Vairappan, et al., 2009). The advantage of a hybrid approach is that it converges much faster, since it reduces the search space dimensions of the back-propagation method used in neural networks.

3.7.1 Architecture of ANFIS

A simple architecture of ANFIS with I-input variables namely $i_1=H_o/L_o$, $i_2=\zeta$, $i_3=B/d$, $i_4=W_{50}/W_{max50}$, $i_5= B/ L_o$ and $i_6= h_B/L_o$ and o is output variable i.e., o=damage level (S)as shown in Fig.3.10. It consists of five layers and each layer is explained below.

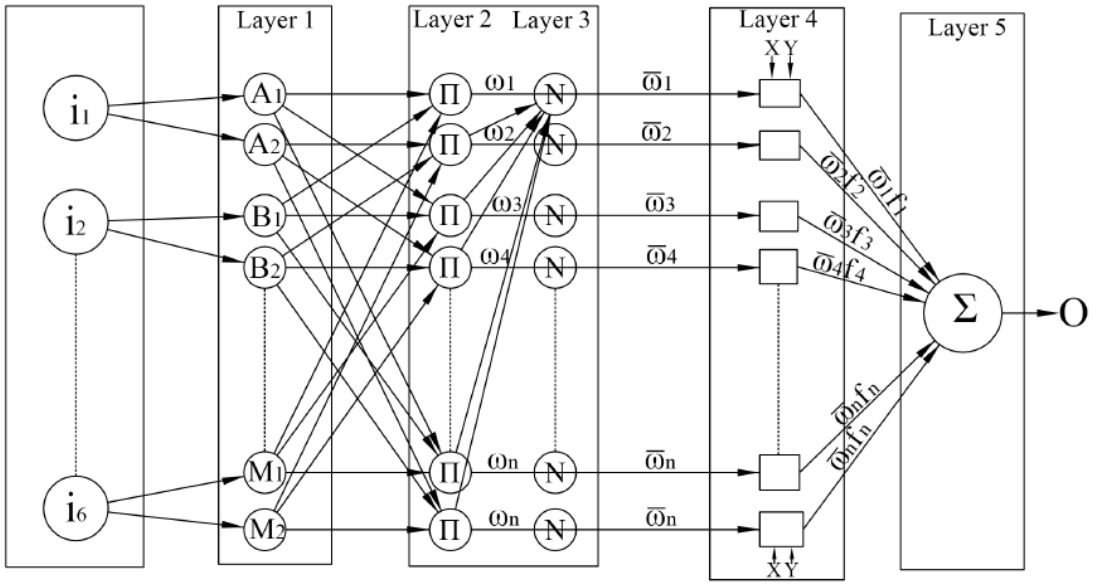


Fig.3.10 ANFIS structure

Layer 1

Every node in this layer is an adaptive node with a node function

$$O_{l,i} = \mu A_i(x), \quad fori = 1,2 \tag{3.23}$$

$$O_{l,i} = \mu B_{i-2}(x), \; fori = 3,4 \tag{3.24}$$

x (or y) is the input node and A_i (or B_{i-2}) is a linguistic variables associated with the membership function of a fuzzy set (A_1,A_2,B_1,B_2).Typical membership function:

$$\mu A(x) = \frac{1}{1+\left[\frac{c_i}{a_i}\right]^{2b_i}} \tag{3.25}$$

where a_i, b_i and c_i is the parameter set. Parameters are called as premise parameters.

Layer 2

Each node in this layer is a fixed node, indicated by Π Norm. The output is the product of all the incoming signals.

$$O_{2,i} = w_i = \mu A_i(x) \cdot \mu B_i(y), \quad i = 1,2 \tag{3.26}$$

output signal w_i represents the fire strength of a rule.

Layer 3

Each node in this layer is a fixed node N Norm. The i^{th} node calculates the ratio of the firing strength to the sum of the firing strength.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2 \tag{3.27}$$

Output signal $\overline{w}_i$ is called normalized firing strengths.

Layer 4

Each node in this layer is an adaptive node, indicated by square node with a node function:

$$O_{4,i} = w_i f_i = w_i(p_i x + q_i y + r_i) \tag{3.28}$$

where w_i is the normalized firing strength from layer 3. $\{p_i, q_i, r_i\}$ is the parameter set which are called as consequent parameters.

Layer 5

Each node in this layer is a fixed node, indicated by circle node which computes the overall output as the summation of all incoming signals:

$$Overalloutput,\ S = O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{w_i} \tag{3.29}$$

The different membership functions assigned for input parameters are:

Gauss membership function:

$$f(x, \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{3.30}$$

where x is input parameters, c and σ are mean and variance respectively.

Triangular membership function:

$$f(x, a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{3.31}$$

where, x is input parameter, a & c locate the feet of the triangle and b locate the peak.

Generalized bell-shaped membership function:

$$f(x, a, b, c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{3.32}$$

where, x is input parameter, c locate the centre of the curve.

3.8 SUMMARY

A detailed physical model study on non-reshaped berm breakwater carried out by (Rao, 2009) is described in this chapter. The wave flume, experimental setup, and experimental procedure of non-reshaped berm breakwater are also discussed. Non-dimensional input parameters that influence the damage level (S) of non-reshaped berm breakwater, such as, deep-water wave steepness (H_o/L_o), surf similarity parameter (ξ), relative berm position (h_B/d), relative armor stone weight (W_{50}/W_{max50}), relative berm width (B/L_o) and relative berm location (h_B/L_o) which are used for the soft computing techniques are discussed. And also the research methodology to develop soft computing tools such as, ANN, SVM, PSO and ANFIS to predict damage level (S). A basic concept of PCA, ANN and SVM which is used in present techniques are explained. Details of proposed PSO based SVM model along with ANFIS are explained in detail.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 GENERAL

For the present study, data was collected from the physical model study on non-reshaped berm breakwater which was carried out in wave flume in marine structures laboratory NITK, Surathkal, India (Balakrishna Rao, 2009). They carried out experiment to study the stability of non-reshaped berm breakwater by taking stone as armor unit. The wave parameters considered for their study was to represent the Mangalore coast. Waves were generated with height varying from 3 m to 4.8 m and period varying from 8 s to 12 s. The depth of water in front of structure was also varied from 9 m to 13. 5 m. The dimensional analysis was carried out on the parameters considered, using Buckingham's-Π theorem. The results thus obtained in dimensionless form was used in the present work to study the performance of soft computing techniques like ANN, SVM and hybrid model ANFIS, PSO-SVM. Methodologies of these techniques were briefly explained in the Chapter 3. Using PCA, influence of input parameters on damage level were carried out. Further the collected data was randomly divided into two set, training about 80% and remaining for testing.

Soft computing techniques such as ANN, ANFIS, SVM and PSO-SVM approach have been used to predict the damage level of the non-reshaped berm breakwater. The capability of the approach was checked by using statistical measures like Correlation Coefficient (CC), Root Mean Square Error (RMSE) and Scatter Index (SI), which are defined as:

$$CC = \frac{\sum_{i=1}^{n}(O_i-\bar{O}_i)(P_i-\bar{P}_i)}{\sqrt{\sum_{i=1}^{n}(O_i-\bar{O}_i)^2(P_i-\bar{P}_i)^2}} \quad (4.1)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \times 100\% \quad (4.2)$$

$$SI = \frac{RMSE}{\bar{O}_i} \quad (4.3)$$

Where, O_i and P_i are observed and predicted damage level respectively, n is the number of data set used and $\bar{O}_i$ and $\bar{P}_i$ are average observed and predicted damage level respectively.

4.2 PERFORMANCE OF ARTIFICAL NEURAL NETWORKS (ANN) MODEL

Seven input parameters namely Armor Stone Weight (W_{50}), Slope angle ($\cot\alpha$), Deepwater Wave Steepness (H_o/L_o), Surf Similarity parameter (ξ), Relative berm position (h_B/d), Relative berm width (B/L_o) and Relative water depth (d/L_o) were considered for this study. Initially, training and testing were carried out using three updated algorithms namely SCG, GDA and LMA for same five number of hidden nodes and a constant 300 epochs in order to arrive at a better training algorithm.

The results obtained during training and testing processes showing CC and mean square error values are tabulated in Table 4.1. The CCs between observed output and network predicted outputs were calculated by using Eq. 4.1. The RMSE and SI between the observed output and network predicted output was calculated using Eqs. 4.2 and 4.3 respectively.

Table 4.1 Statistical results obtained for ANN1 Model

Network training function		CC	RMSE	SI
GDA	Train	0.717	4.96	0.416
	Test	0.711	4.99	0.375
SCG	Train	0.933	2.57	0.215
	Test	0.834	3.94	0.296
LMA	Train	0.945	2.33	0.195
	Test	0.888	3.47	0.260

From Table 4.1, it can be observed that the CC obtained after GDA training for ANN1 model was 0.717 and for testing the model it was found that test CC was 0.711. The RMSE value was found to be 4.96 & 4.99 for train and test data. SI was 0.416 and 0.375 for test and train data. The same ANN1 model was trained by SCG which showed a correlation of 0.933 for training and for testing a correlation of 0.834.The RMSE and SI in this case was found to be 2.57 , 3.94 0.215 & 0.296 for train and test data respectively. The ANN1 model again with trained by LMA was found was found to have a correlation of 0.945 and 0.888 for training and testing respectively. The RMSE and SI for the ANN1 model with LMA were found to be 2.33 & 3.47 and 0.195 & 0.260 for train and test data respectively. It can be observed clearly that ANN1 model with LMA showed a better correlation compared to the other two algorithms and also the RMSE was found minimum. Final CC's for training and testing for GDA, SCG and LMA are shown in Figs. 4.1 to 4.3. From the results shown in Table 4.1 it can be observed that the CC and RMSE were better for LMA algorithm compared to SCG and GDA. Hence, further studies were carried out using LMA.

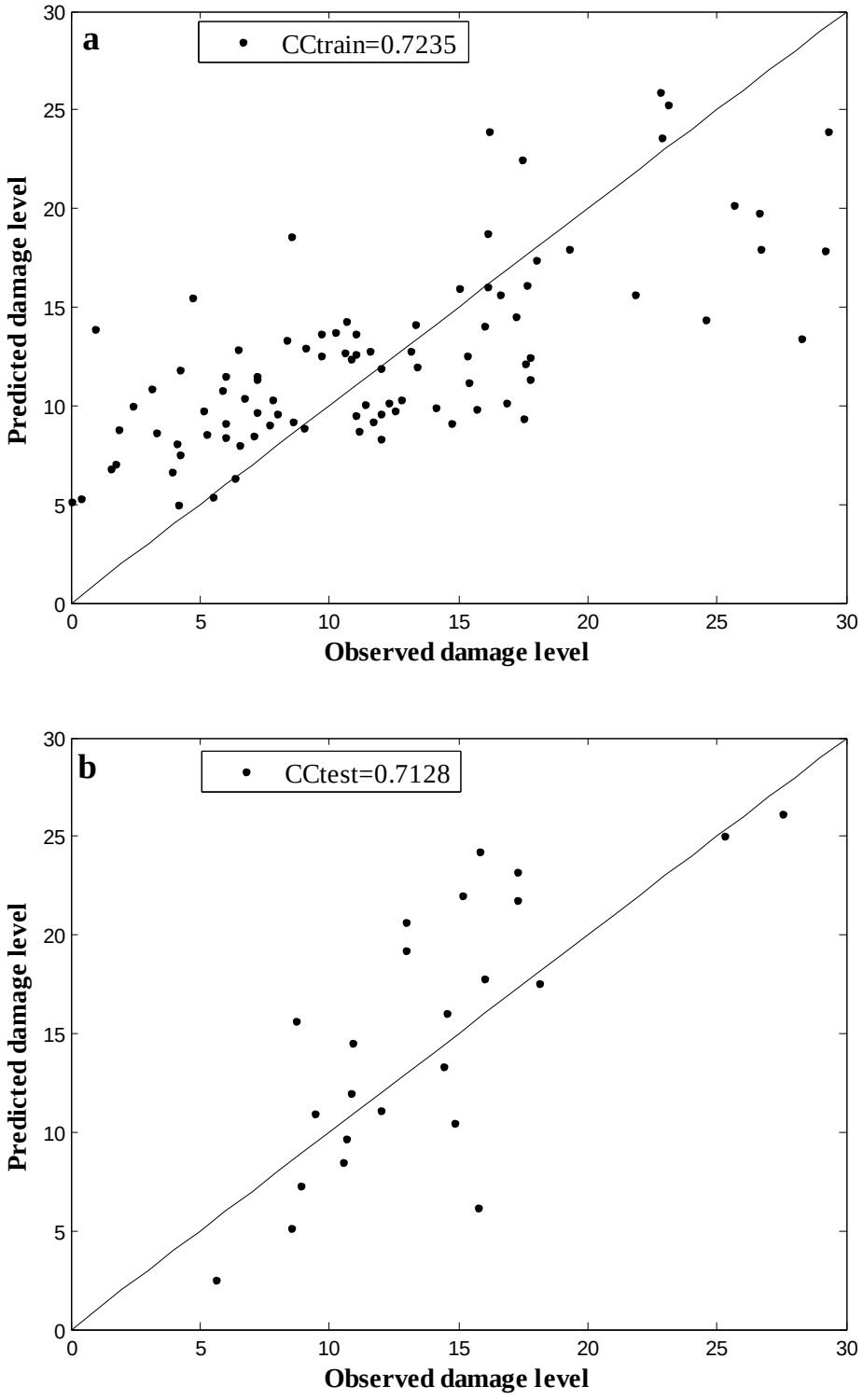


Fig 4.1 Observed and predicted damage level for by ANN1 (GDA) model for (a) CC_{train} and (b) CC_{test}

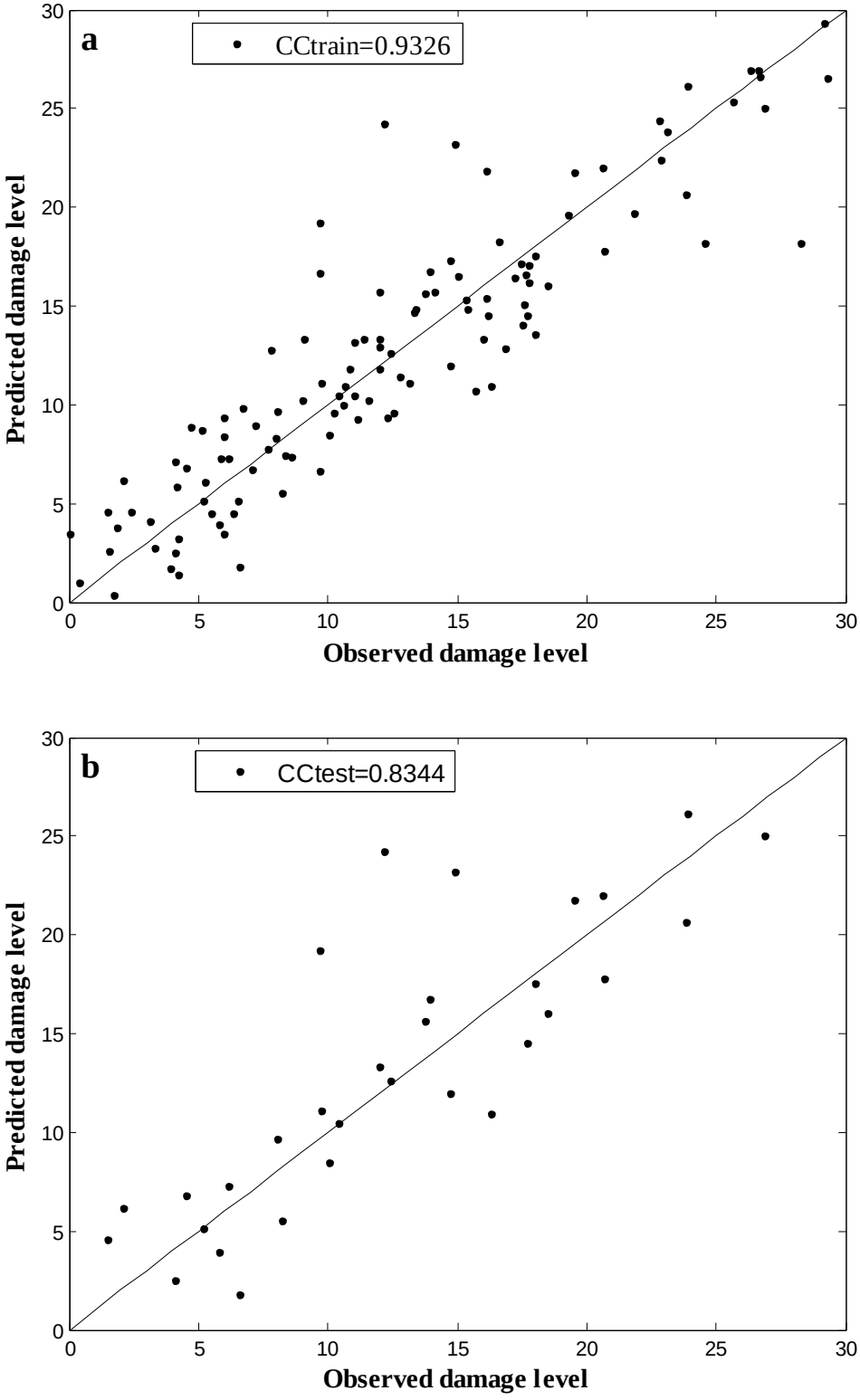


Fig 4.2 Observed and predicted damage level for by ANN1 (SCG) model for (a) CC_{train} and (b) CC_{test}

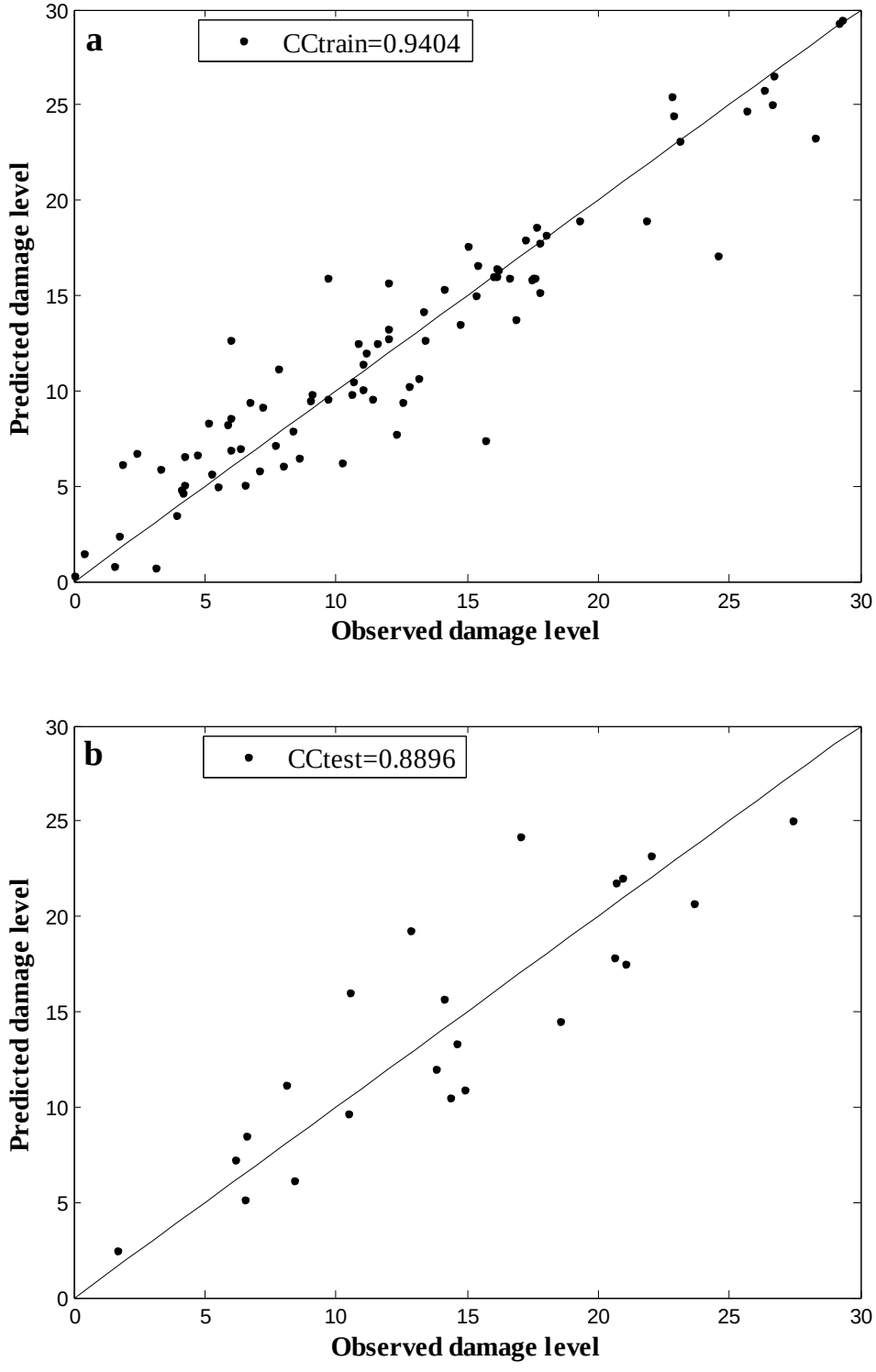


Fig 4.3 Observed and predicted damage level for by ANN1 (LMA) model for (a) CC_{train} and (b) CC_{test}

4.3 PRINCIPAL COMPONENT ANALYSIS

PC analysis was carried out using statistiXL software (www.statistixl.com/downloads/files). Analysis part consisted of calculation of principal components for the given set of data. In the present case, seven parameters were considered for analysis namely Armor Stone Weight (W_{50}), Slope angle ($\cot\alpha$), Wave Steepness (Ho/Lo), Surf Similarity parameters (ζ), Relative berm width by water depth (B/d), Relative berm width (B/ Lo) and Relative berm location (h_B/Lo). The principal components were calculated for all these parameters. The results of the PC analysis are tabulated in Tables 4.2 and 4.3.

Table 4.2 Eigen values for all PCA and percentage of variance

Value	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7
Eigen value	2.955	1.927	1.358	0.343	0.241	0.106	0.069
% of Var.	42.221	27.532	19.400	4.904	3.447	1.510	0.986
Cum. %	42.221	69.753	89.153	94.057	97.504	99.014	100.000

Table 4.3 Principal Component loadings

Variable	**PC 1**	**PC 2**	**PC 3**
Armor Stone Weight (W_{50})	0.088	0.118	**-0.949**
Slope ($\cot\alpha$)	0.464	-0.626	-0.515
Wave Steepness (H_o/L_o)	**0.922**	0.243	0.193
Surf Similarity (ζ)	**-0.944**	0.001	0.097
Relative berm width by water depth (B/d)	-0.400	**0.757**	-0.368
Relative berm width (B/ L_o)	0.094	**0.892**	-0.011
Relative berm location (h_b/L_o)	**0.907**	0.305	0.099

It can be observed in Table 4.2 that the Eigen values and the percentage of variance are greater than one and 10% respectively for PC1, PC2 and PC3. Hence, further PCA loading analysis was carried out for these three principal components. The components with loading greater than or equal to 75% were considered to be influencing the output i.e., damage level "S". The results of the loadings calculated

for the three principal components are shown in Table 4.3 which clearly shows that the "cotα" parameter has a loading less than 75% and hence the influence of that parameter on damage level was less. Henceforth, six input parameters was considered for the further analysis using various soft computing techniques. Table 4.4 shows the new input parameters obtained after PC analysis along with their percentage of influence.

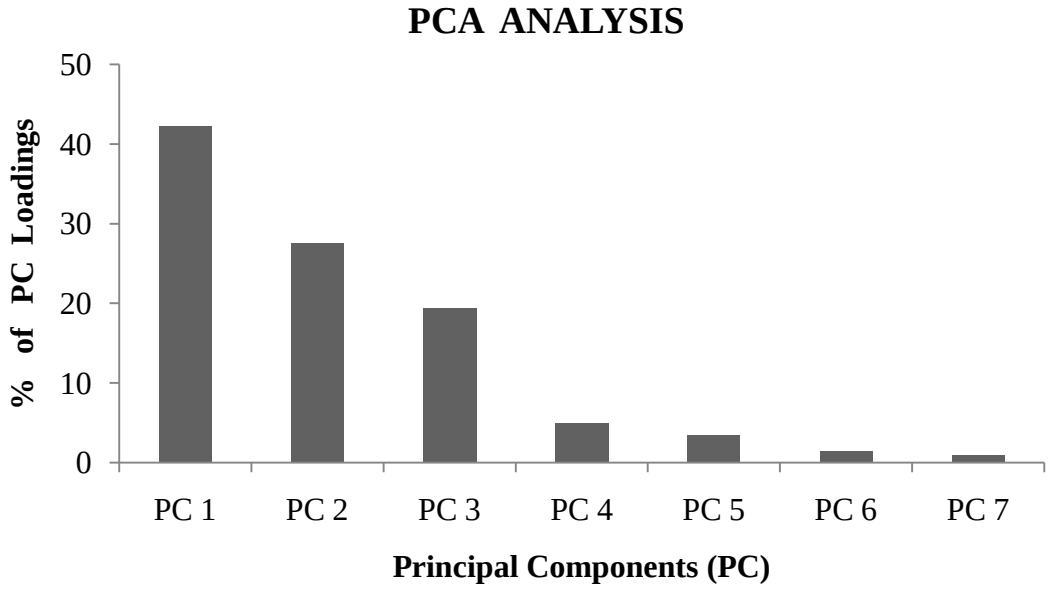


Fig. 4.4 Principal components v/s percentage of PC loadings

Table 4.4 New input parameters after PC analysis

Sl. No.	Parameters	Percentage of influence
1	Armor stone weight	0.949
2	Surf Similarity	0.944
3	Wave Steepness	0.922
4	Relative berm location	0.907
5	Relative berm width	0.892
6	Relative berm width by water depth	0.757

4.4 PERFORMANCE OF FEED FORWARD BACK-PROPAGATED NEURAL NETWORK MODEL FOR REDUCED INPUT PARAMETER

Results of the analysis carried out with seven input parameters was explained in previous section 4.2 and in this section the discussion of analysis carried out with reduced parameters obtained after the PC analysis was explained.

The reduced network model with six input parameters was constructed and trained by LMA. Table 4.5 shows the results in terms of CC, RMSE and SI values obtained during training and testing processes with different network combinations. A total of 300 epochs were carried out during the analysis for each combination of network. The CC value was found to be same as network analysis carried out with seven input parameters. Hence, it can be concluded that six input parameters are sufficient for the present study.

Table 4.5 Statistical measures obtained for ANN2 Model with different network combinations

Network		CC	RMSE	SI
6-3-1	Train	0.873	3.417	0.283
	Test	0.805	4.371	0.337
6-4-1	Train	0.908	2.935	0.243
	Test	0.854	3.947	0.303
6-5-1	Train	0.940	2.388	0.198
	Test	0.884	3.687	0.283
6-6-1	Train	0.935	2.486	0.206
	Test	0.865	3.757	0.288
6-7-1	Train	0.929	2.592	0.215
	Test	0.844	4.590	0.352

Further, LMA was used for network analysis of damage level prediction with varying hidden nodes from 3 to 7 for 300 epochs. Here the CC's of predicted values increase

with increase in number of hidden nodes. Also, with the increase in number of hidden nodes the error between observed output and predicted output was found to be decreasing. But, after certain number of increase in hidden layers, the error again starts increasing.

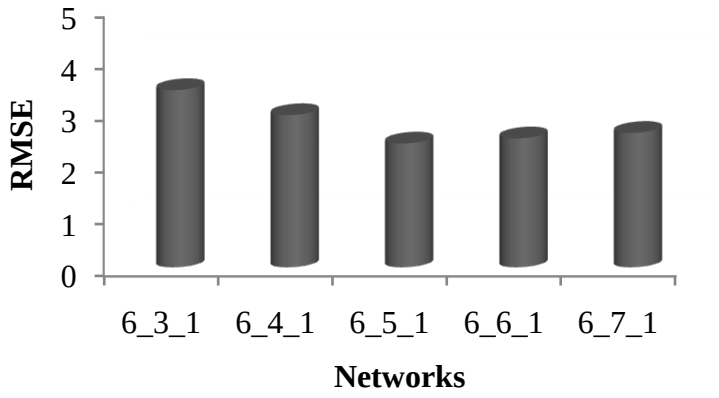


Fig. 4.5 RMSE for each network for train data

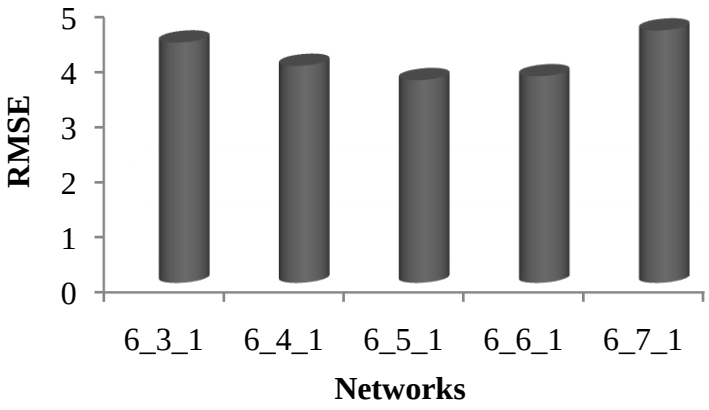


Fig. 4.6 RMSE for each network for test data

Mirchandani and Cao (1989) and Huang and Huang (1991) have suggested that the number of hidden layers may be equal to one less than the number of input parameters to increase the learning efficiency. Considering these criteria and Figs. 4.5 and 4.6 in the present case, it was observed that 5 numbers of hidden layers gave least error with high CC. A CC of 0.94 for training set and 0.88 for testing set with a RMSE of 2.388 for training and 3.687 for testing sets and a lower SI of 0.198 and 0.283 for training and testing sets was achieved with 5 hidden nodes and 300 epochs with the considered

network models. The CC graph for 6-5-1 network of training and testing was plotted as shown in Fig 4.7.

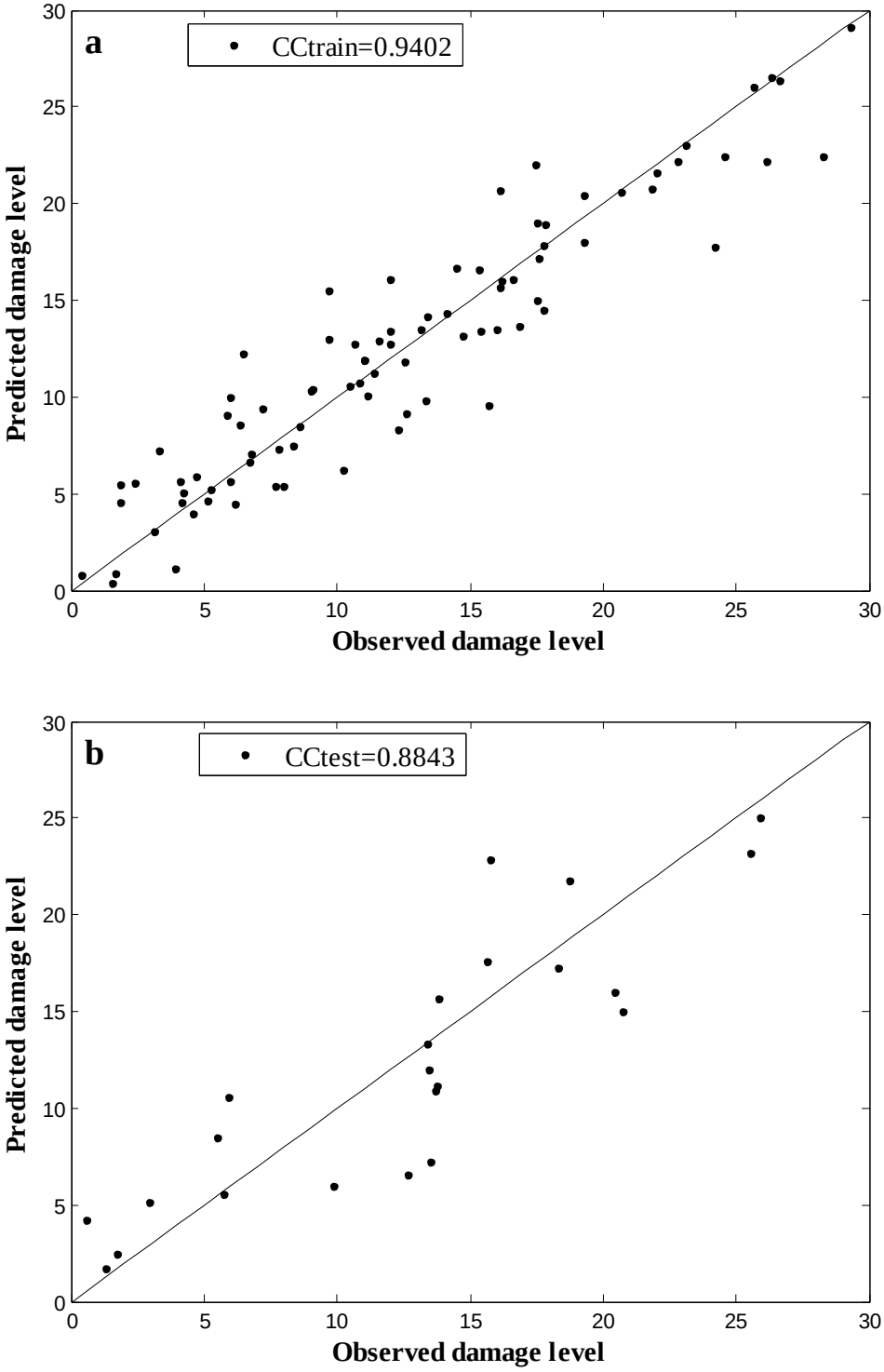


Fig. 4.7 Observed and predicted damage level for by ANN2 (LM) model for (a) CC_{train} and (b) CC_{test}

4.4.1 Estimation of Damage Level by ANN2 Model

Once the network was trained the weight and bias values got fixed for that model. The ANN2 structure constructed for estimating damage level of non-reshaped berm breakwater was shown in Fig. 4.8. The structure consists of six input nodes, five hidden layer nodes, and one output node. After training the network model, weights and biases of the network are fixed. These fixed weight and bias values are shown in Fig. 4.8.

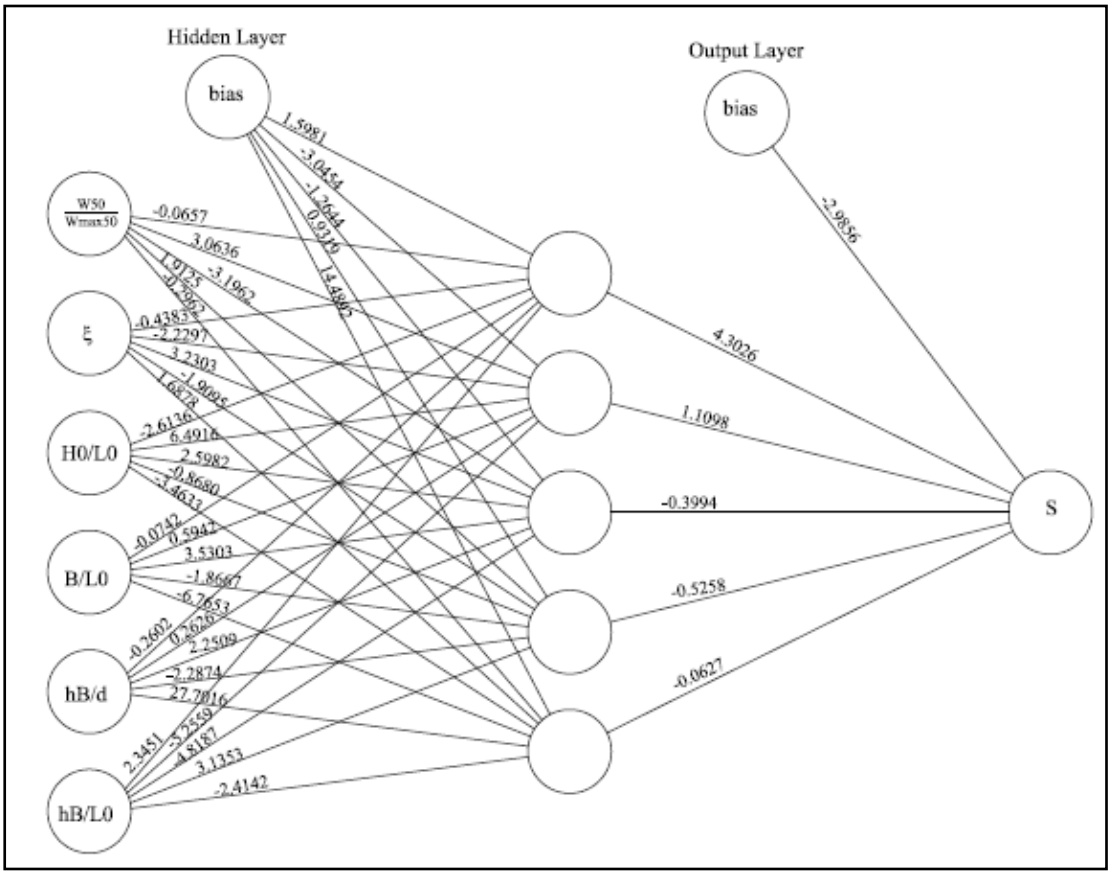


Fig. 4.8 ANN2 structure with weights and biases

Here each input value was multiplied with the weight and adds with bias value, total sum was then the input at each hidden layer node, which was passed through a hyperbolic tangent sigmoid transfer function as defined in equation 4.4. Further the output from hidden node get multiplied with the weight and adds with the bias value, and then the total sum was passed through purelin as shown in equation 4.5. The damage level (S) was estimated using following formulations:

Hyperbolic tangent sigmoid transfer function $F_{ih} = \left[\frac{2}{1+\exp(-2\times N_{ih})} - 1\right]$ (4.4)

N_{ih} are values of input to hidden nodes and F_{ih} are the Hyperbolic tangent sigmoid transfer functions of hidden node ih. where $ih =$ 1 to 5,

$$N_1 = W_{50}/W\max_{50}(-0.0657) + H_0/L_0(0.4383) + \xi(-2.6136) + B/d(-0.0742) + B/L_0(-0.2602) + h_B/L_0(2.3451) + 1.5981$$

$$N_2 = W_{50}/W\max_{50}(3.0636) + H_0/L_0(-2.2297) + \xi(6.4916) + B/d(0.5942) + B/L_0(0.2626) + h_B/L_0(-5.2559) - 3.0454$$

$$N_3 = W_{50}/W\max_{50}(-3.1962) + H_0/L_0(3.2303) + \xi(2.5982) + B/d(3.5303) + B/L_0(2.2509) + h_B/L_0(-4.8187) - 1.2644$$

$$N_4 = W_{50}/W\max_{50}(1.9125) + H_0/L_0(-1.9095) + \xi(-0.8680) + B/d(-1.8669) + B/L_0(-2.2874) + h_B/L_0(3.1353) + 0.9319$$

$$N_5 = W_{50}/W\max_{50}(-0.2960) + H_0/L_0(1.6878) + \xi(-3.4633) + B/d(-6.7653) + B/L_0(27.7016) + h_B/L_0(-2.4142) + 14.492$$

Above equations are transferred to hyperbolic sigmoid transfer functions (Eq. 4.4).

$$F_1 = [2/1+\exp(-2\times N_1)] - 1$$

$$F_2 = [2/1+\exp(-2\times N_2)] - 1$$

$$F_3 = [2/1+\exp(-2\times N_3)] - 1$$

$$F_4 = [2/1+\exp(-2\times N_4)] - 1$$

$$F_5 = [2/1+\exp(-2\times N_5)] - 1$$

N_1 to N_5 and F_1 to $F5$, represents summation function and transfer function at each hidden node respectively.

$$N_{ho} = F_1(0.68521) + F_2(4.3026) + F_3(1.1098) + F_4(-0.3994) + F_5(-0.5258) + F_6(-0.0627) - 2.9856$$

Above equation was transferred to purelin transfer functions

Purelin transfer function $G_{ho} = purelin(N_{ho})$ (4.5)

N_{ho} are values of hidden to output nodes and G_{ho} is purelin transfer function of output node ho.

Then Damage $S = purlin(G_{ho})$ (4.6)

Equation 4.6 provides trained ANN2 model for estimating damage level of non-reshaped berm breakwater.

It is observed that the maximum R obtained during the tests was 0.884 and there are chances of further improving the solution. Also, with each repetition of the ANN model the results vary. This is because ANNs uses empirical risk minimization and follows a heuristic path, with applications and extensive experimentation preceding theory. Further, ANNs can suffer from multiple local minima which lead to over fitting.

In contrast, SVMs use structural risk minimization which involves sound theory first, then implementation and experiments. SVMs solution is global and unique. It has a simple geometric interpretation and gives a sparse solution. Unlike ANNs, the computational complexity of SVMs does not depend on the dimensionality of the input space. The reason that SVMs often outperform ANNs in practice is that they deal with the biggest problem with ANNs, SVMs are less prone to over fitting. Because of all these advantages of SVM, it was decided to carry out the further tests using SVM.

4.5 PERFORMANCE OF SUPPORT VECTOR MACHINE (SVM) MODEL

The performance of SVM depends on the good setting of SVM and kernel parameters. In developing SVM models, initially parameters are randomly selected by coarse search (i.e. for C = 100,200, 300….2000; ε = 0.5, 1…2; γ = 1,2,…6 and d = 1,2,...6) to identify the near optimal values, and then a fine search (i.e. for C=1,10,20, 30….2000; ε = 0.000001,…2; γ = 0.01,0.02,…6 and d = 1,2....6) was carried out to identify the final optimal values. The final optimum values of SVM and kernel parameters are shown in Table.4.6.

Table.4.6 Optimal parameters for SVM models with different kernel functions

Kernel	nsv	C	ε	γ	d
Linear	86	1260	0.00126	-	-
Polynomial	86	415	0.00035	-	3
rbf	86	300	0.00030	1	-
erbf	86	20	0.00002	6	-
Spline	86	1	0.000001	-	-
Bspline	86	10	0.00001	-	1

The statistical parameters computed using the predicted and observed damage level of training and testing data for the SVM models are shown in Table.4.7. The number of support vectors for all the SVM models are 100%, which indicates that there was no noise in the data set. The SVM model with linear kernel function results low CC (Training CC = 0.3590, Testing CC = 0.3583) and RMSE, SI are very high when compared to other kernel function in terms of statistical measures. The SVM model with bspline kernel function over-fit the data, as we get very high training accuracy and a much lower testing accuracy when compared to SVM models with kernel functions polynomial, rbf, erbf, spline, as shown in Table.4.7.

Table.4.7 Statistical measures of SVM models

Kernel functions of SVM Model	Training CC	Testing CC	Training Data		Testing Data	
			RMSE	SI	RMSE	SI
Linear	0.3590	0.3583	6.0746	0.5360	6.3190	0.5222
Polynomial	0.9081	0.8883	2.8811	0.2542	3.2357	0.2674
rbf	0.8795	0.8665	3.1259	0.2758	3.4569	0.2857
erbf	0.7824	0.7697	4.5877	0.4048	4.9127	0.4060
Spline	0.8983	0.8609	2.9335	0.2589	3.4851	0.2880
Bspline	0.9970	0.6845	0.5074	0.0448	5.8227	0.4812

The better selection of SVM and kernel parameters decides the performance of these models. In case of rbf and erbf kernel, the optimal width (γ) obtained by manual search are found to be 1 and 6 respectively. The optimal values of d (degree) in case

of polynomial and bspline kernel function obtained by manual search are 3 and 1 respectively. The SVM model with polynomial kernel function shows better generalization performance with CC 0.9081 and 0.8883, RMSE 2.8811 and 3.2357, SI 0.2542 and 0.2674 for training and testing respectively when compared to all other kernel functions. The CC of SVM models for the training and testing data are shown in Fig. 4.9 – 4.14.

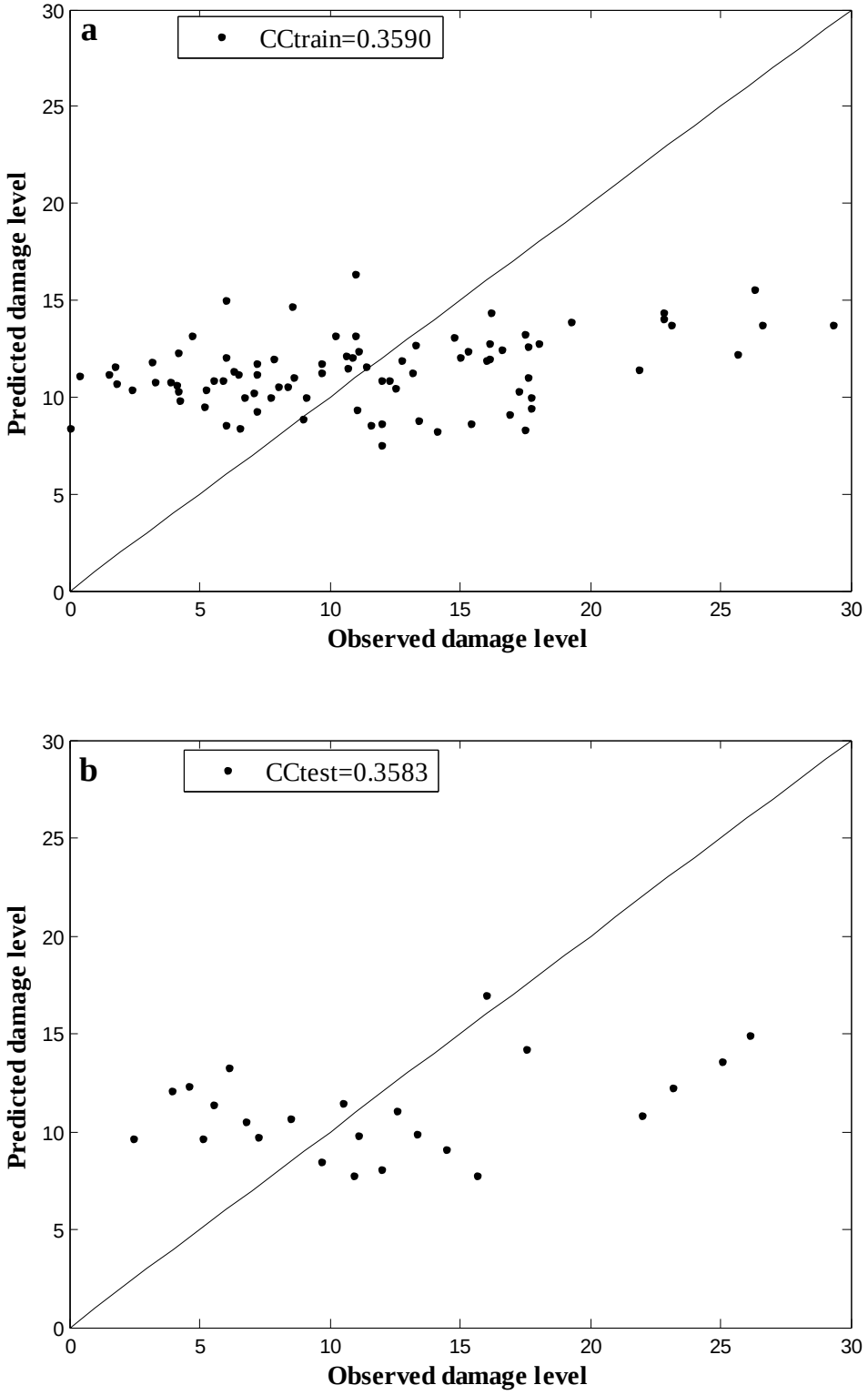


Fig 4.9 Observed and predicted damage level for linear kernel using SVM for (a) CC_{train} and (b)CC_{test}

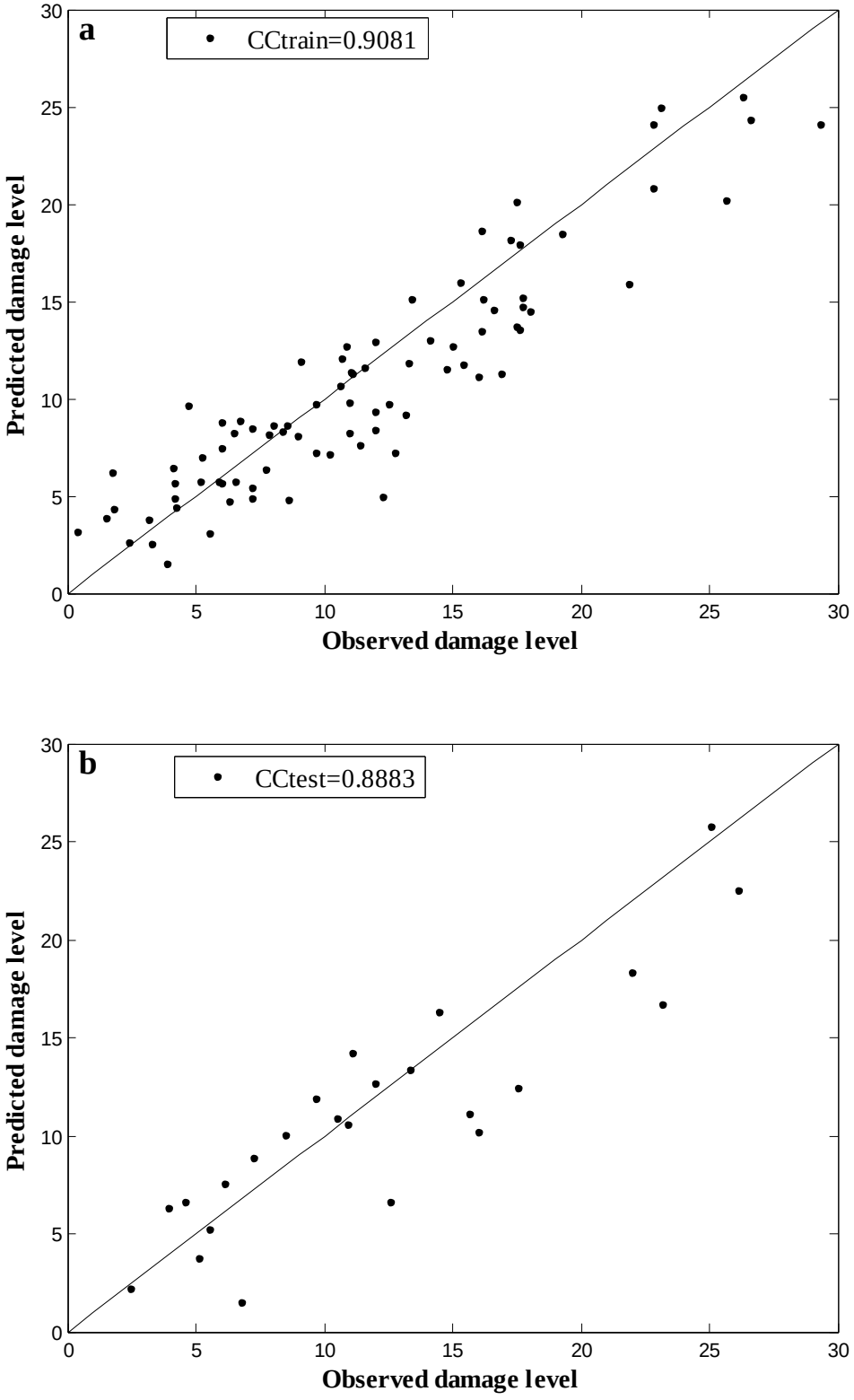


Fig 4.10 Observed and predicted damage level for polynomial kernel using SVM for (a)CC_{train} and (b)CC_{test}

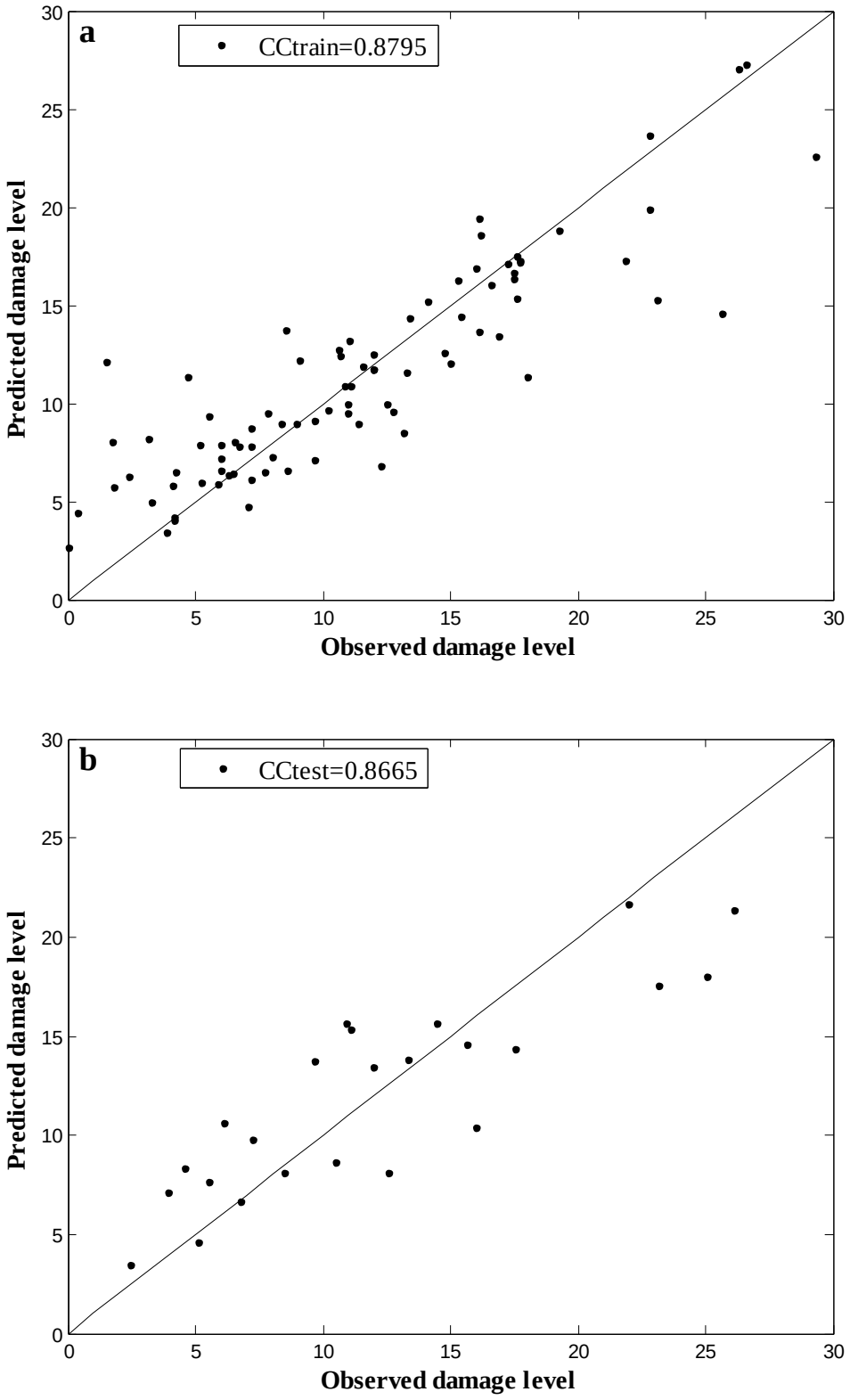


Fig 4.11 Observed and predicted damage level for rbf kernel using SVM for (a) CC_{train} and (b)CC_{test}

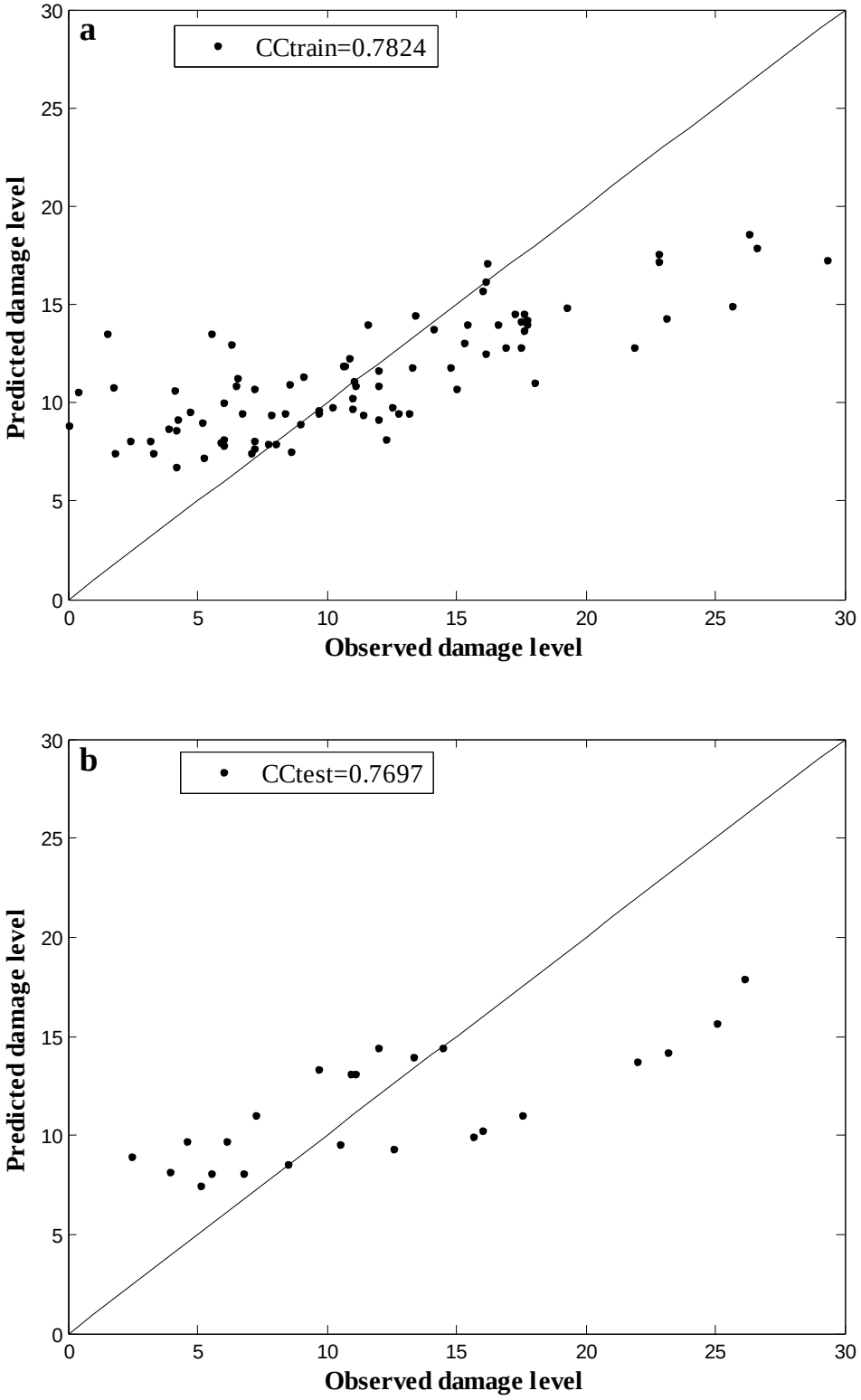
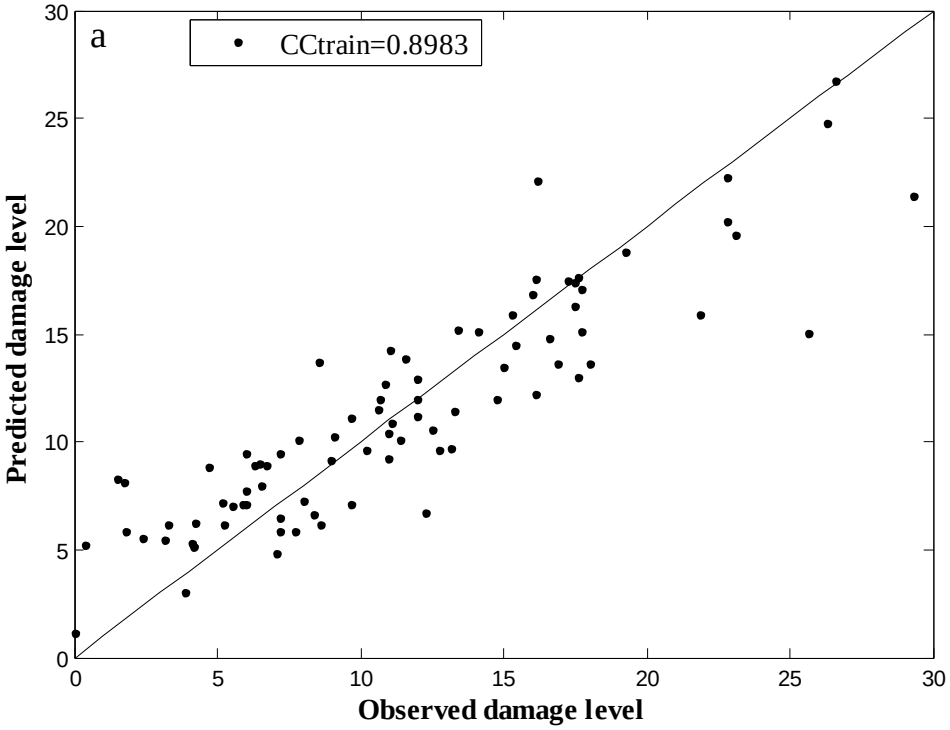


Fig 4.12 Observed and predicted damage level for erbf kernel using SVM for (a) CC_{train} and (b)CC_{test}



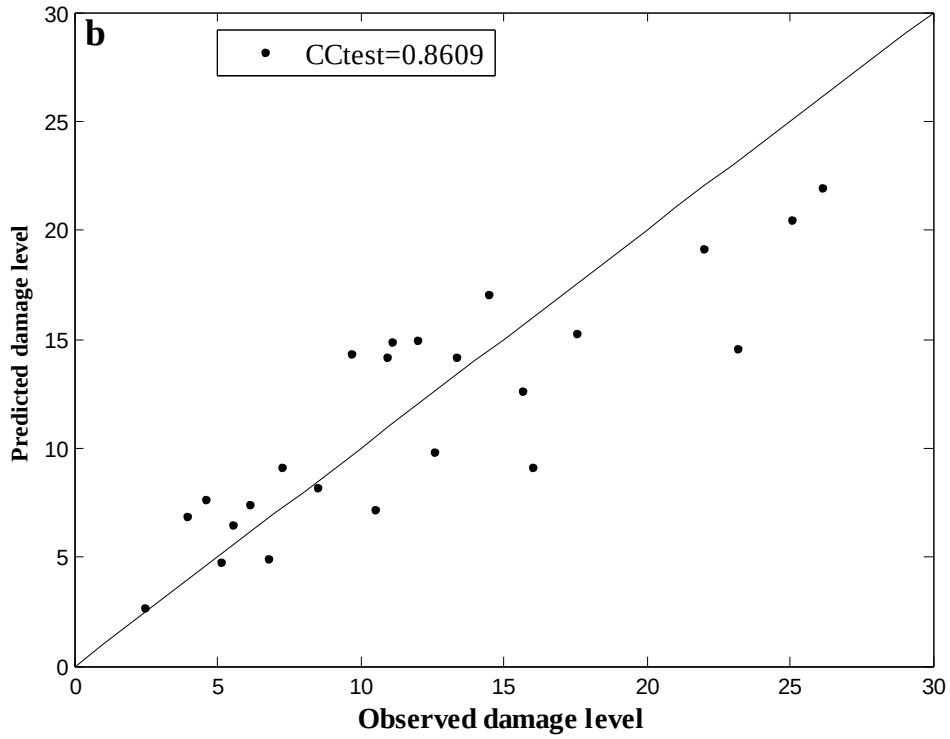
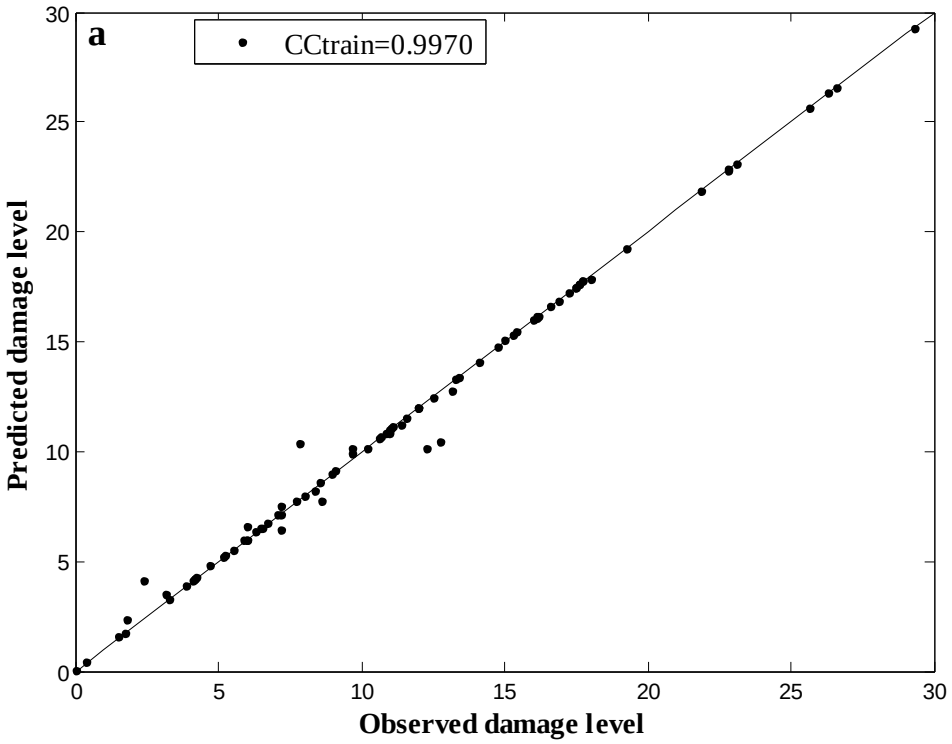


Fig 4.13 Observed and predicted damage level for spline kernel using SVM for (a) CC_{train} and (b)CC_{test}



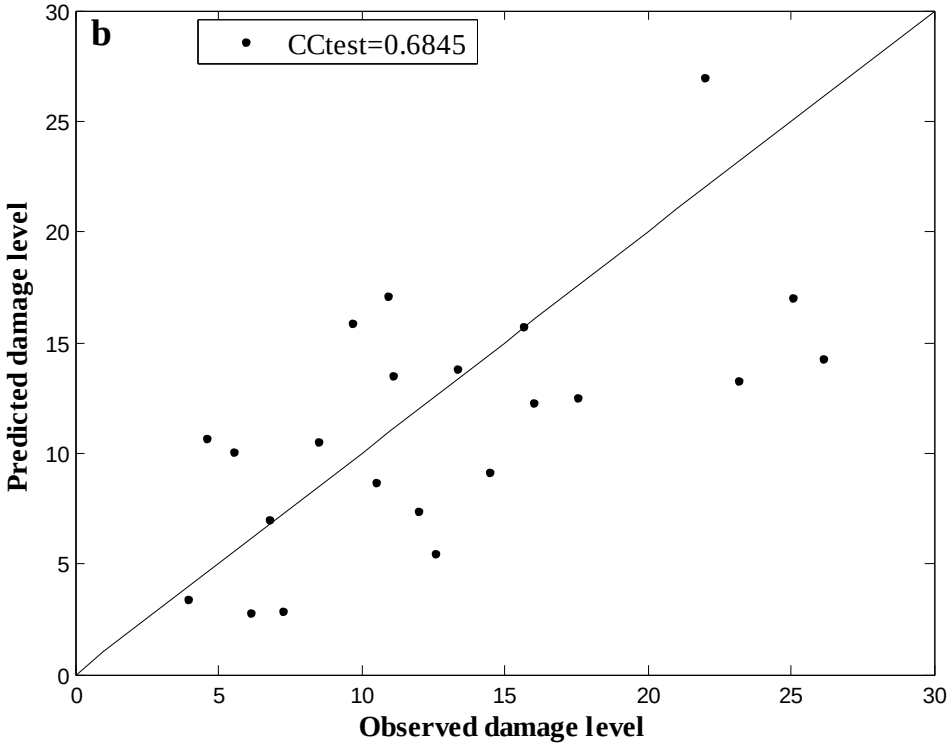


Fig 4.14 Observed and predicted damage level for bspline kernel using SVM for (a) CC_{train} and (b)CC_{test}

4.6 COMPARISON OF ANN AND SVM MODELS

Results obtained from both the models are shown in Table 4.8. Statistical measures of both the models gave satisfactory results. But, in comparison to ANN, SVM results were better and also SVM gave a unique solution with any number of trails if it was run where as ANN results were varying as the weights and bias changed during each trail run of the model.

Table 4.8 Comparison of ANN and SVM statistical measures

Model	Training CC	Testing CC	Training Data		Testing Data	
			RMSE	SI	RMSE	SI
ANN (6-5-1)	0.940	0.884	2.388	0.198	3.687	0.283
SVM (Polynomial	0.9081	0.888	2.8811	0.2542	3.2357	0.2674

By observing results as shown in Table 4.8 the CC's obtained for both models for testing are 0.884 and 0.888 which was less in soft computing techniques. This can be further improved by hybridizing these models with other techniques (Fuzzy, PSO etc.,). Further, from the literature it was also observed that hybrid models were better than the single models (Patil et al. 2012). To improve the results, hybridizing technique such as ANN with Fuzzy and PSO with SVM were implemented and analyses were carried out. The obtained results are discussed in the next few sections.

4.7 PERFORMANCE OF PARTICLE SWARM OPTIMIZATION (PSO) BASED SUPPORT VECTOR MACHINE (SVM) MODEL

While using PSO-SVM model in the training stage for damage level prediction the kernel parameters (d, γ) and SVM parameters (C, ε) have to be optimized by PSO and the error has to be measured using Eq. 4.3. The kernel parameters with minimum error namely (d, γ) and SVM parameters were considered for the further studies. The testing data sets were used to examine the accuracy of the prediction. The parameters of PSO were set as follows: the size of the population as 30, set learning factors C1 = 0.2 and C2 = 2.5 and the inertia weight *'w'* linearly increased from 0.4 to 0.9.

Out of six SVM models only four were considered for this hybrid model. The linear and bspline (Table 4.7) were ignored since it showed poor generalization performance. The SVM models with bspline kernel functions over-fit the data, which show a very high training accuracy and a much lower testing accuracy. PSO was applied to remaining four models. SVM and kernel parameters obtained by PSO-SVM are shown in Table.4.9. Statistical measures computed using the predicted and observed damage level of training and testing data for the four PSO-SVM models are shown in Table.4.10.

Table 4.9 Optimal parameters for PSO-SVM models with different kernel functions

Kernel	nsv	C	ε	γ	D
		PSO-SVM	PSO-SVM	PSO-SVM	PSO-SVM
Polynomial	86	4.55	0.0000455	-	3.9
rbf	86	45899	0.045899	1.1	-
erbf	86	193860	0.19386	14	-
Spline	86	6.612	0.0000066	-	-

Table 4.10 Statistical measures of PSO-SVM models

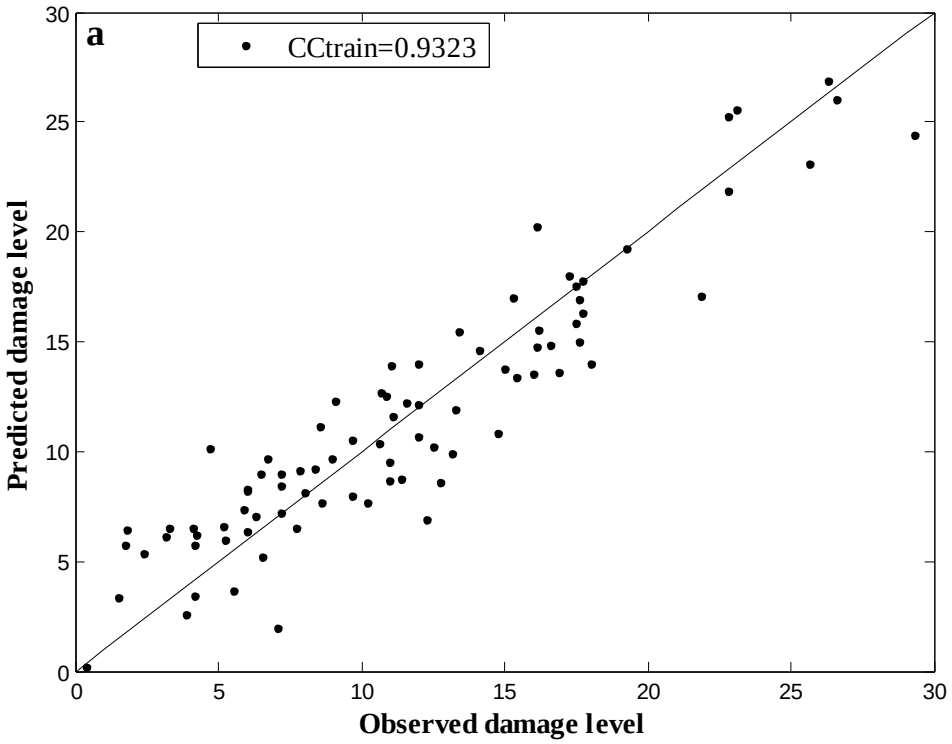
Kernel functions of SVM Model	Training CC	Testing CC	Training Data		Testing Data	
			RMSE	SI	RMSE	SI
Polynomial	0.9323	0.9207	2.3500	0.2074	2.6230	0.2168
rbf	0.8956	0.8742	3.2994	0.2911	3.7311	0.3083
erbf	0.8675	0.7700	3.6936	0.3259	4.6045	0.3805
Spline	0.9555	0.8770	1.9592	0.1729	3.2640	0.2640

In comparison to all PSO-SVM models, the model accompanying erbf kernel function shows less generalization performance (CC Train = 0.8675 and CC Test = 0.7700) in prediction of damage level of statically stable non-reshaped berm breakwater with RMSE 3.6936 and 4.6045, SI 0.3259 and 0.3805 for training and testing data respectively when compared to all other kernel functions. The number of support

vectors used in PSO-SVM models was 100% (86), which indicates that every training data set was utilized as support vector. This clearly proves that, there was no noise in the training data set, but there was non-linearity and complexity associated in mapping input and output parameters of non-reshaped berm breakwater. Increasing C will disturb the solution, In case of rbf and erbf kernel, the values of regularization parameter C obtained by PSO are found to be 45899 and 193860 respectively. The optimal values of C in case of polynomial and spline kernel function obtained by PSO are 4.55 and 6.612 respectively. The PSO-SVM model with polynomial kernel function shows better generalization performance with CC 0.9207, RMSE 2.6230, and SI 0.2168 for the testing data when compared to all other kernel functions.

The performance of PSO-SVM model with spline kernel function (CC Train = 0.9555 and CC Test = 0.8770) is better than PSO-SVM models with rbf and erbf kernel functions in terms of statistical measures RMSE 1.9592 and 3.2640, SI 0.1729 and 0.2640 for training and testing data respectively.

It was noticed that the performance of these models depends on the better selection of SVM and kernel parameters. The kernel and SVM parameters obtained by PSO (Table.4.9) is tested by using test data set, which shows better generalization performance with highest Training CC = 0.9207 for PSO-SVM model with polynomial kernel function. By interfacing PSO with SVM, generalization performance of the PSO-SVM models shows improvement in terms of statistical measures such as CC, RMSE, and SI over SVM models. This can be observed comparing Table.4.7 and Table.4.10. The CC of four PSO-SVM models for the training and testing data are shown in the Fig. 4.15 - 4.18.



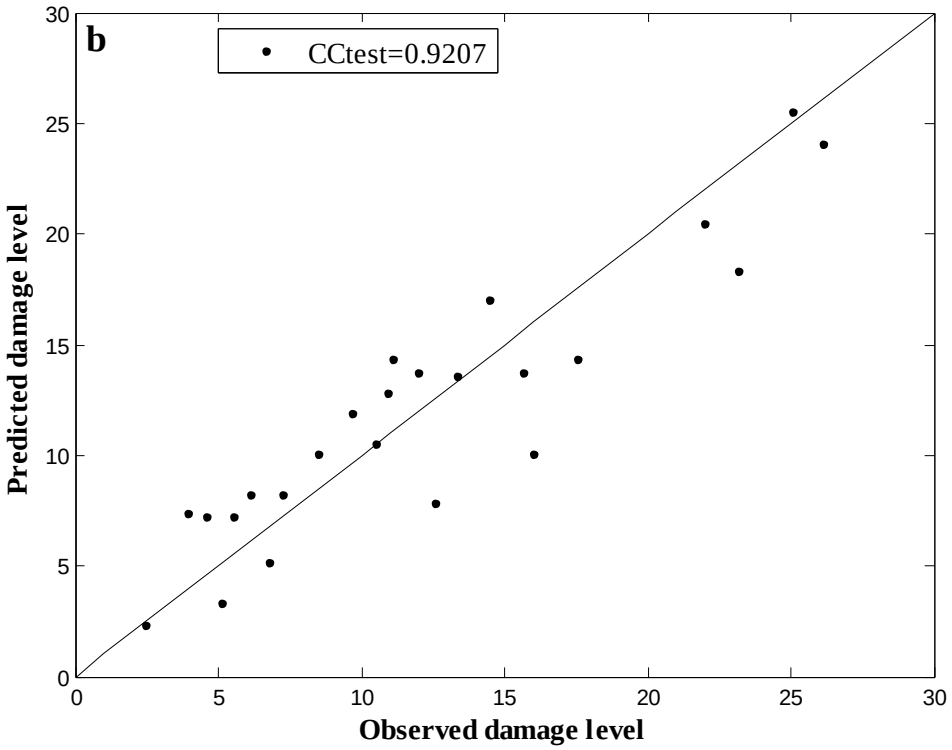


Fig 4.15 Observed and predicted damage level for polynomial kernel using PSO-SVM for (a) CC_{train} and (b) CC_{test}

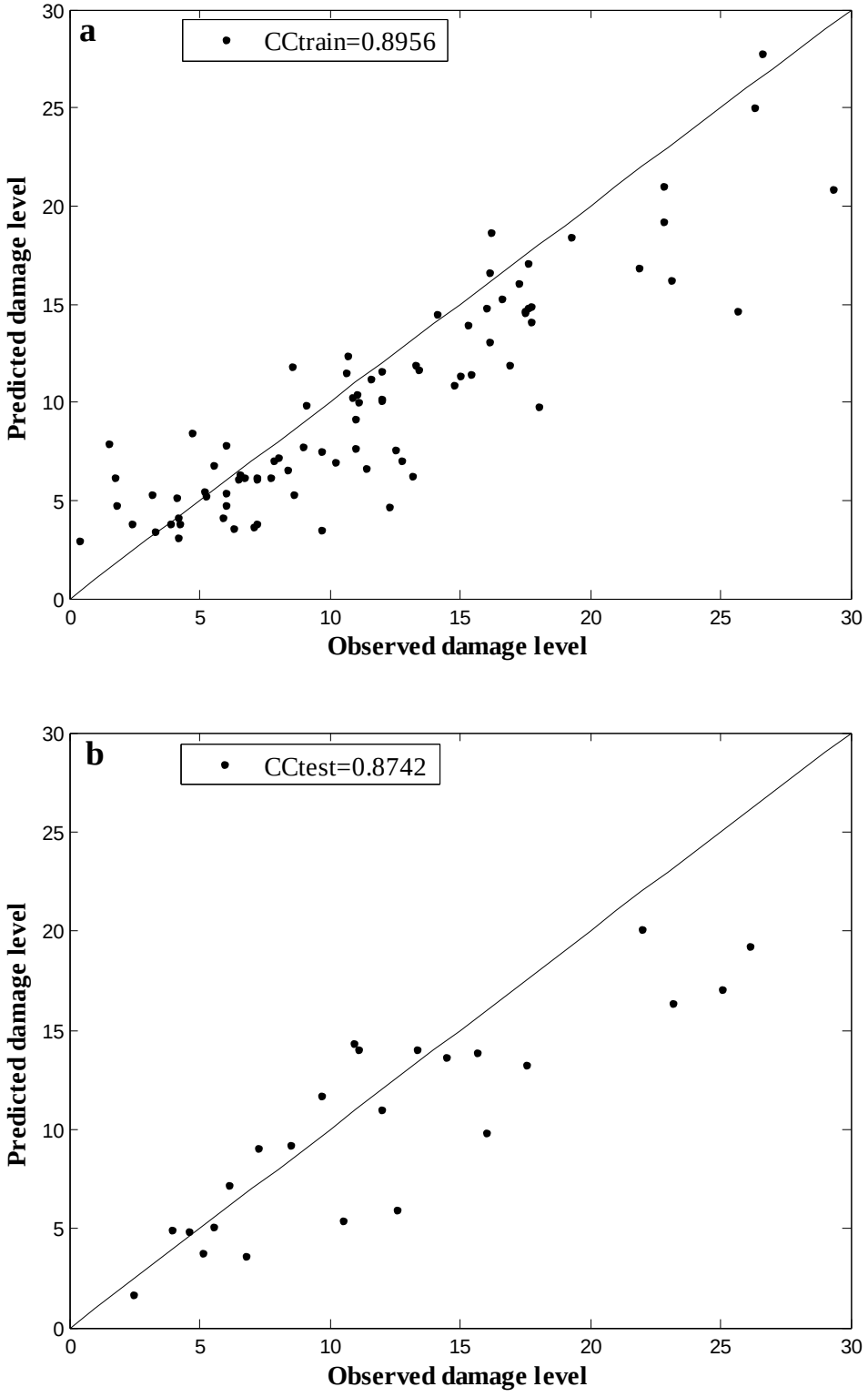
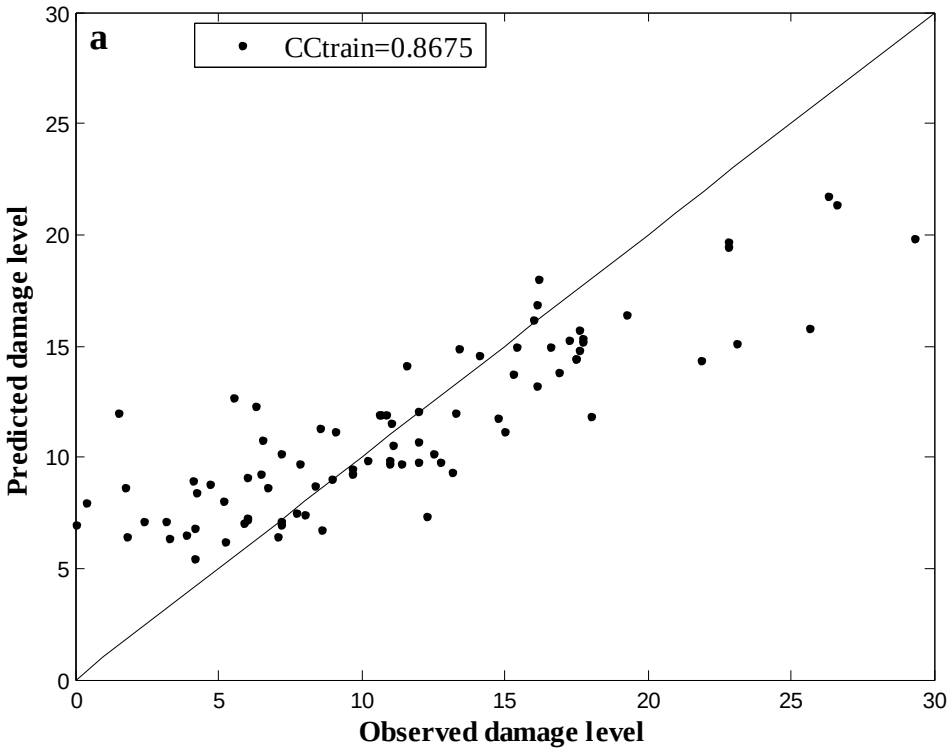


Fig 4.16 Observed and predicted damage level for rbf kernel using PSO-SVM for (a) CC_{train} and (b)CC_{test}



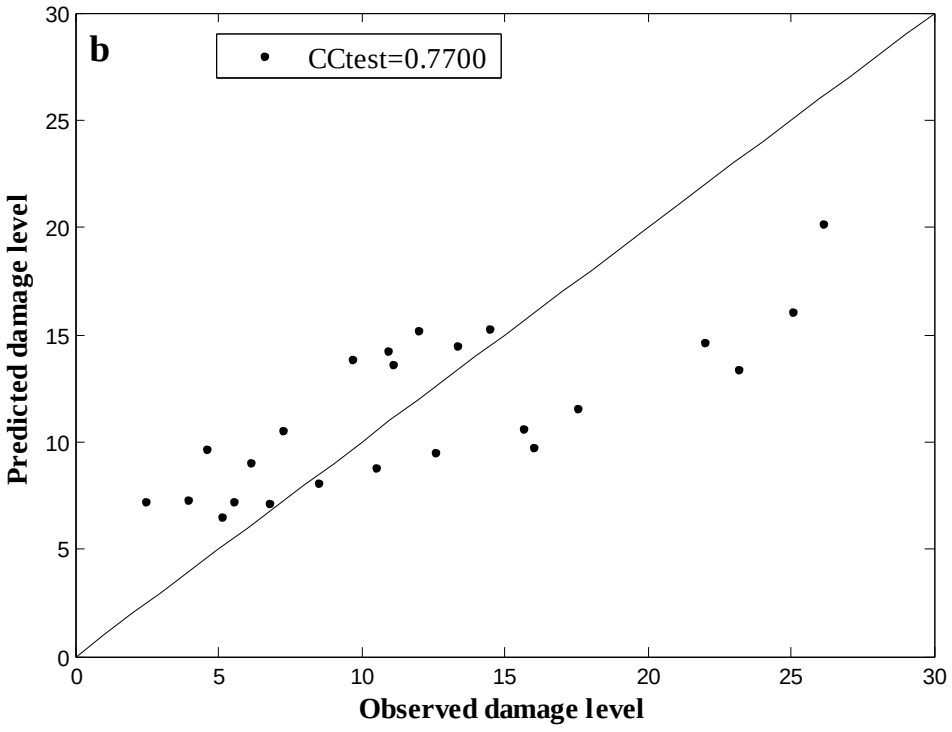


Fig 4.17 Observed and predicted damage level for erbf kernel using PSO-SVM for (a) CC_{train} and (b)CC_{test}

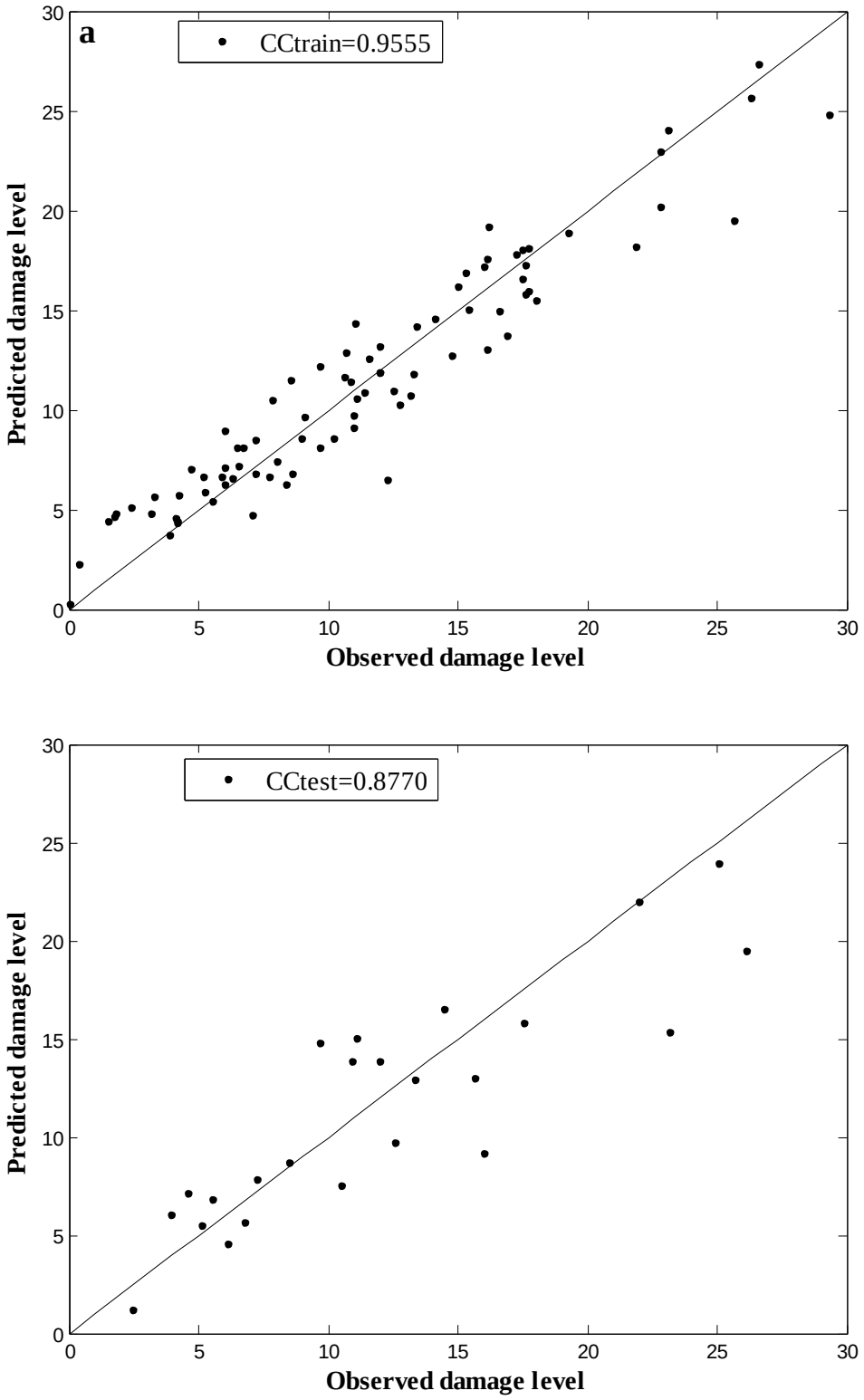


Fig 4.18 Observed and predicted damage level for spline kernel using PSO-SVM for (a) CC_{train} and (b) CC_{test}

4.8 PERFORMANCE OF ADAPTIVE NEURO FUZZY INFERENCE SYSTEM (ANFIS) MODEL

The ANFIS model with six input parameters and one output parameter was constructed with different membership functions, fuzzy rules and epoch numbers. The various membership functions considered were Triangular-shaped built-in membership function (TRIMF), Trapezoidal-shaped built-in membership function (TRAPMF), Generalized bell-shaped built-in membership function (GBELLMF), and Gaussian curve built-in membership function (GAUSSMF). An ANFIS model with built-in 2 membership functions for each variable, and 100 epoch numbers were selected and trained using the Levenberg–Marquardt algorithm with tangent sigmoid and linear transfer functions in the hidden and output layers, respectively.

The results obtained during training and testing processes showing CC and RMSE values are tabulated in Table 4.11. The CC between desired output and network predicted outputs were calculated by using Eq. 4.1. The RMSE and SI between target output and network predicted output is calculated by using Eqs. 4.2 and 4.3 respectively. The correlation coefficient graph for the ANFIS model considering all the membership functions are shown in Fig. 4.19 to 4.22.

Table 4.11 Statistical measures for ANFIS model

Membership Function		RMSE	CC	SI
Triangular	Train	0.143	0.997	0.0123
	Test	5.028	0.764	0.4155
Trapezoidal	Train	1.227	0.985	0.102
	Test	7.451	0.662	0.572
Gbell	Train	0.123	0.998	0.0109
	Test	3.146	0.861	0.260
Gauss	Train	0.147	0.997	0.013
	Test	2.145	0.938	0.177

From Table 4.11 it was observed that the CC obtained after training by TRIMF for ANFIS model was 0.997 and for testing it was 0.764 with a RMSE of 0.143 and 5.028

for train and test data. Further, with TRAPMF a CC of 0.985 and 0.662 was obtained for training and testing with a RMSE of 1.227 & 7.451 for train and test data respectively. The same ANFIS model trained by GBELLMF showed a CC of 0.998 and 0.861 for training and testing with a RMSE of 0.123 & 3.146 for train and test data. The ANFIS model trained with GAUSSMF showed a CC of 0.997 and 0.938 for training and testing data with a RMSE of 0.147 & 2.145 for train and test data respectively. From the tabulated results and discussion it was clear that the ANFIS model using GAUSSMF was better suited for the present data set with high CC values and low RMSE values.

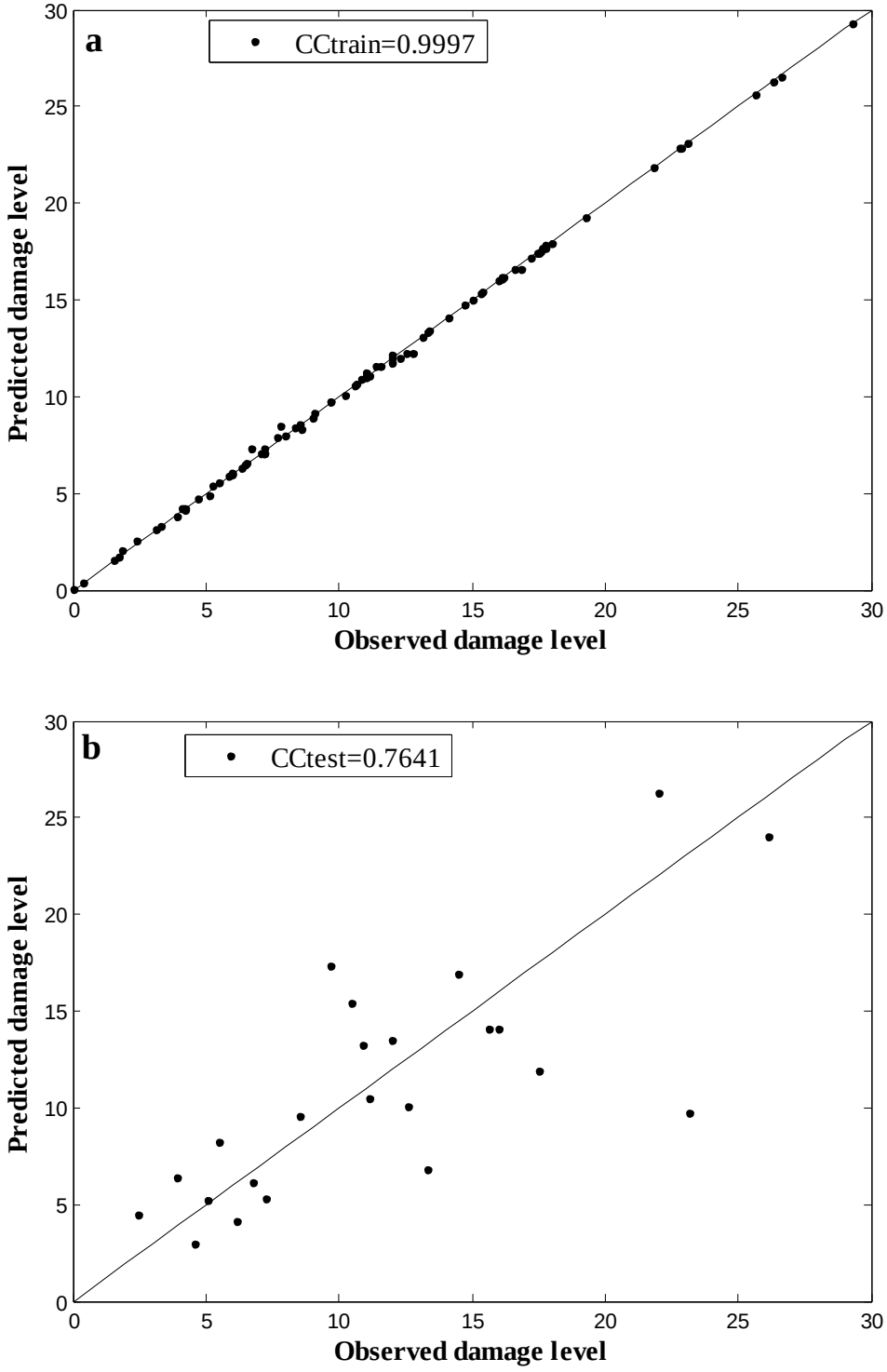


Fig. 4.19 Observed and Predicted damage level by ANFIS (Triangular) model for CC_{train} and CC_{test} data

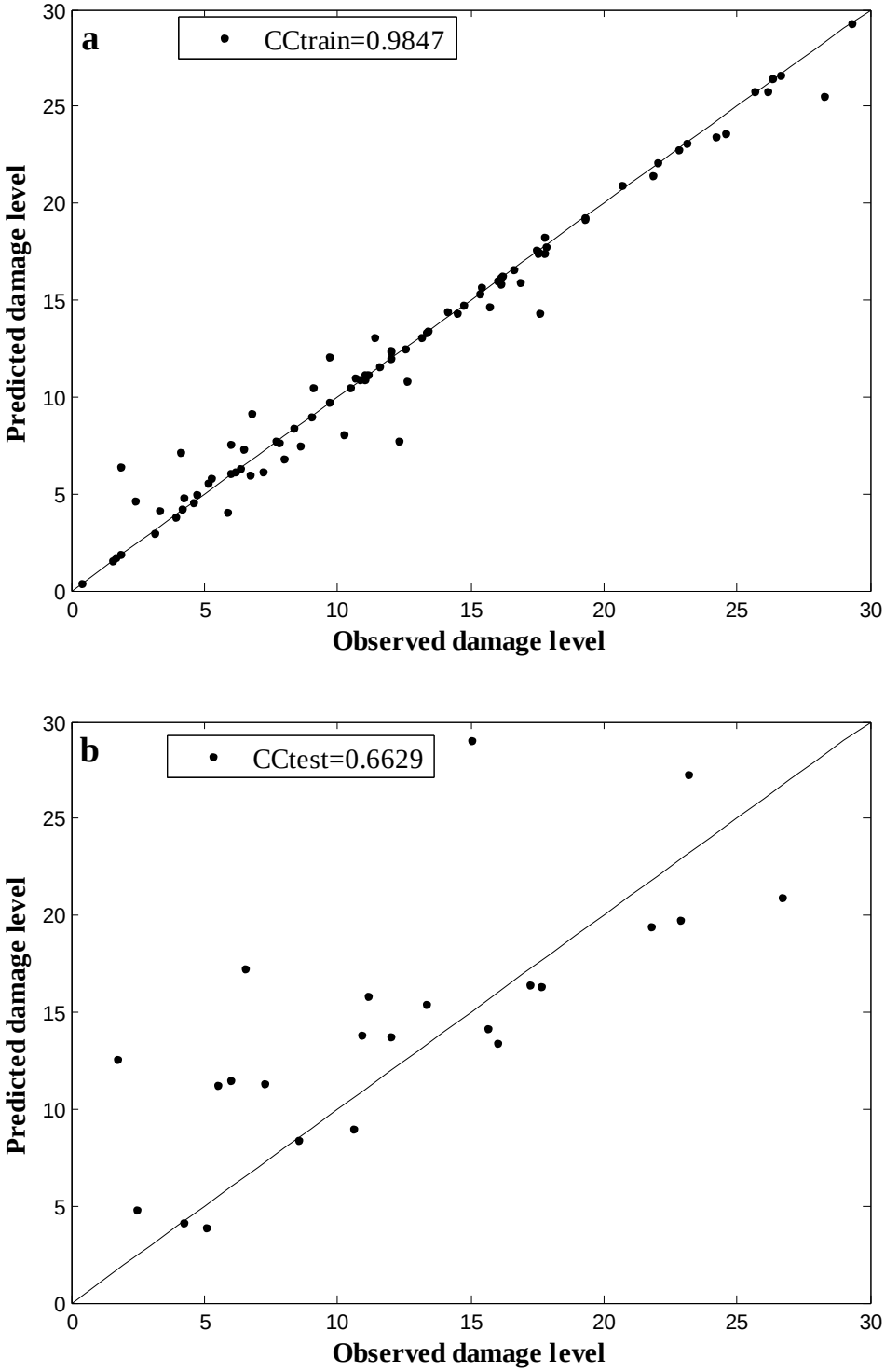


Fig. 4.20 Observed and Predicted damage level by ANFIS (Trapezoidal) model for CC_{train} and CC_{tes} data

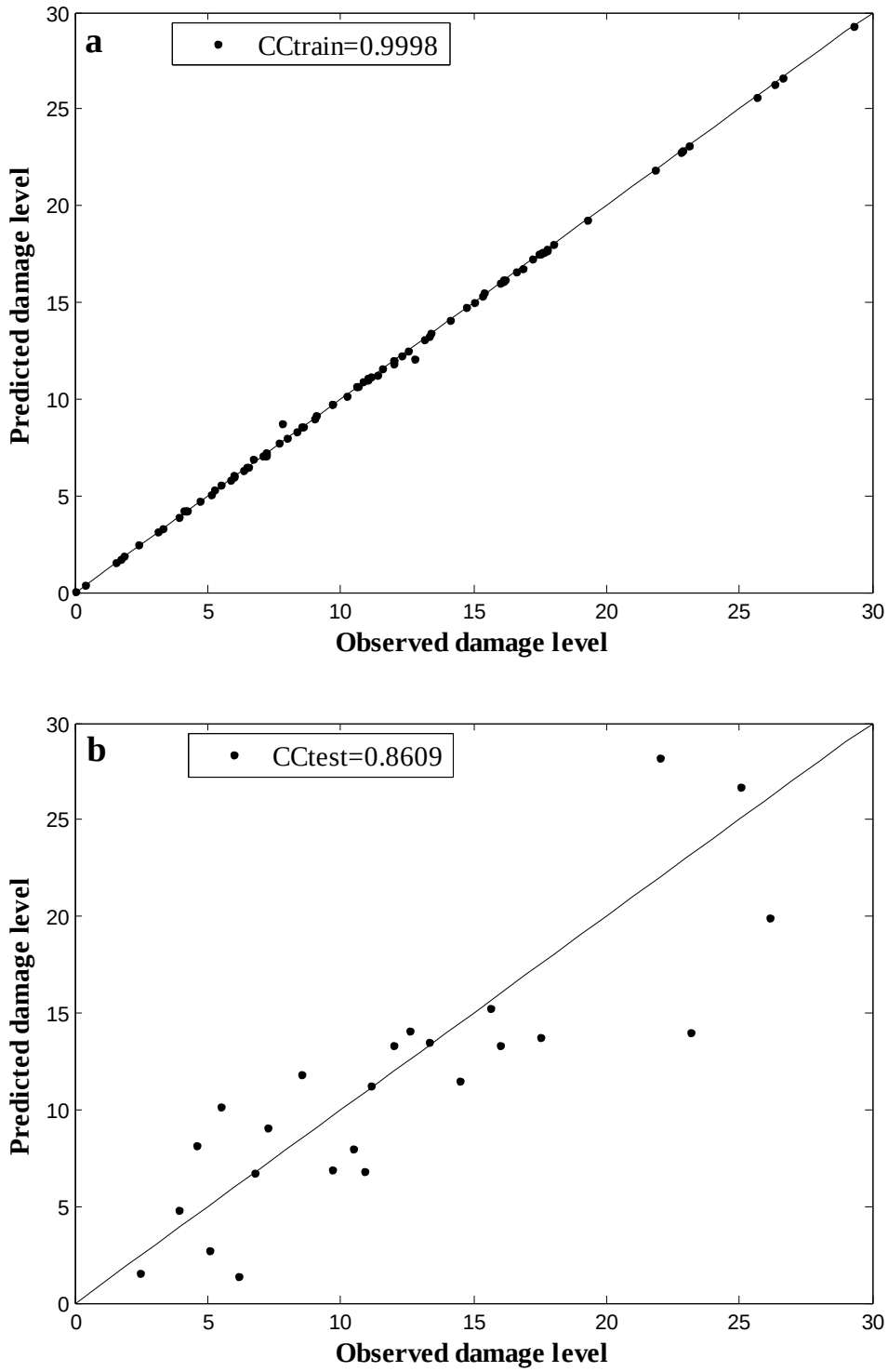


Fig. 4.21 Observed and Predicted damage level by ANFIS (Gbell) model for CC_{train} and CC_{test} data

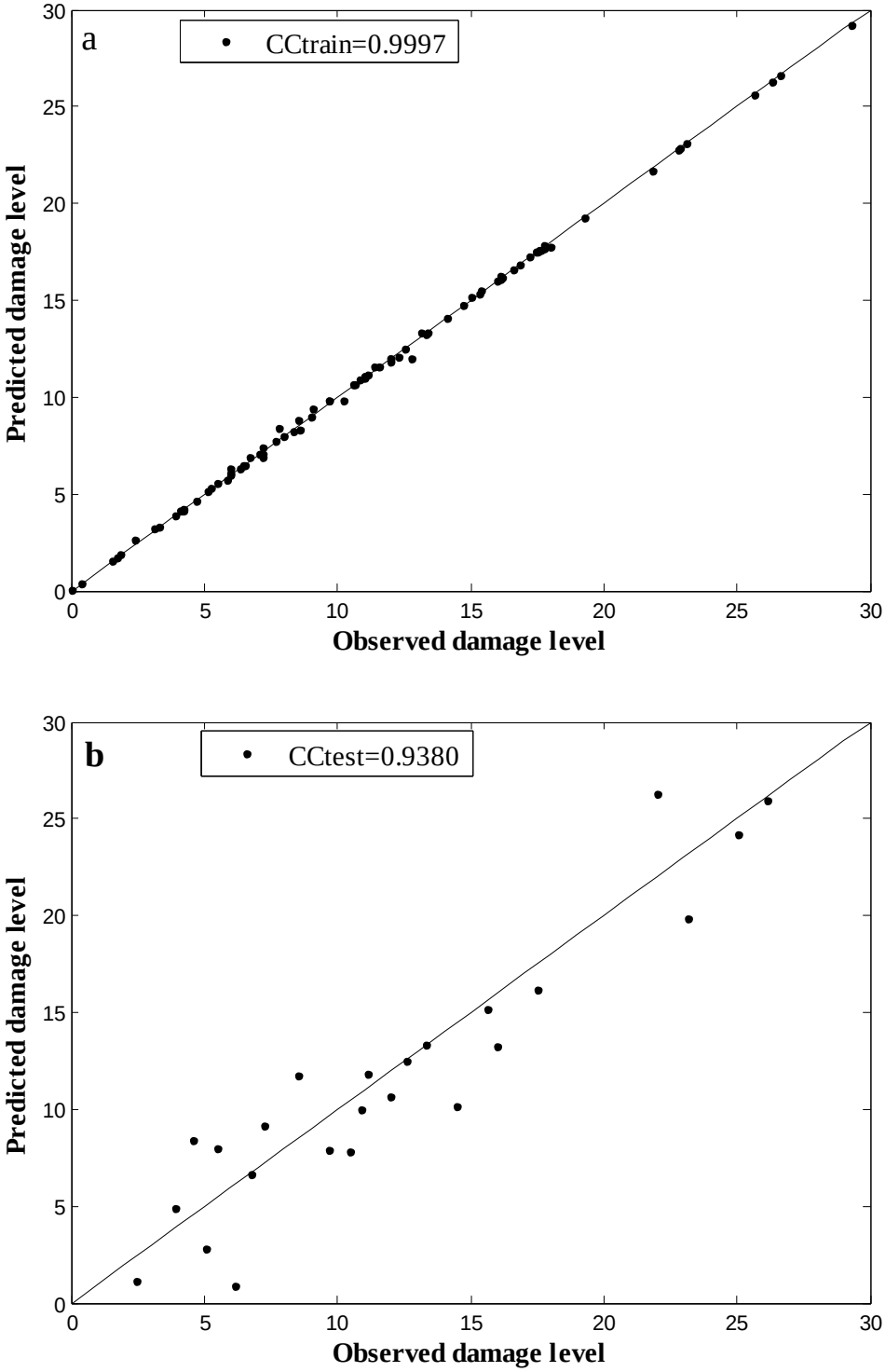


Fig. 4.22 Observed and Predicted damage level by ANFIS (Gauss) model for CC_{train} and CC_{test} data

4.9 PERFORMANCE EVALUATION OF ANN, SVM, ANFIS AND PSO-SVM MODEL

The results obtained using individual models have been discussed in previous sections. In this section a comparison of all the models developed has been done to know the best model that could predict damage level of non-reshaped berm breakwater precisely and accurately. From Tables 4.5, 4.7, 4.10 and 4.11 it was observed that the individual models with their particular kernel functions or algorithms showed better results. Considering the models with their best kernel function or algorithm the present comparison study was carried out and the results are tabulated in Table 4.12.

Table.4.12 Comparing Statistical Measures of ANN, SVM, ANFIS and PSO-SVM models

Model	Training CC	Testing CC	Training Data		Testing Data		Computing Time Secs
			RMSE	SI	RMSE	SI	
ANN (LMA)	0.940	0.884	2.388	0.198	3.687	0.283	1.56
SVM (Polynomial)	0.9081	0.888	2.8811	0.2542	3.2357	0.2674	14.67
PSO-SVM (Polynomial)	0.9323	0.921	2.3500	0.2074	2.6230	0.2168	19.45
ANFIS (GaussMF)	0.997	0.938	0.147	0.013	2.145	0.177	25.87

In terms of hybrid model comparison, ANFIS model performed better than the individual ANN model and PSO-SVM model with polynomial kernel function showed better results than the SVM model with polynomial kernel function in terms of statistical measures. Comparing all the models it was clear from the table that the ANFIS model with gaussMF showed better performance than all the other models

considered in the study with a CC of 0.997 and 0.938 for training and testing data respectively.

From Table 4.12, comparison of ANN, SVM, ANFIS and PSO-SVM models with statistical measures and computational time of each model. All the models were run in DELL INSPIRON with Intel® core ™ i5 CPU @ 2.67 GHz and 4 GB RAM and 64 bit windows 7 operating system. It was observed that the hybrid models in both cases showed better results compared to individual models. When the hybrid models were compared, ANFIS model gave higher CC and lower RMSE. But considering computational time ANFIS has taken more time then PSO-SVM model say around 25% extra computational time. Hence PSO-SVM was computationally efficient as compared to ANFIS.

4.10 SUMMARY

Initially, PC analysis was also carried out to study the influence of individual input parameters on the output. The PC analysis carried out showed that six input parameters were more influencing the output which was considered for the next studies. The number of hidden layers and epochs were optimized further before using them in the model. In ANN it was observed that 5 hidden layers with 300 epochs were sufficient to get better results from the different models. Firstly, the individual models of ANN and SVM were tested with six input parameters and with different algorithms and kernel functions respectively to know the influence of each in the respective models. It was observed that ANN with LM algorithm and SVM with Polynomial kernel function performed better compared to other algorithms and kernel functions.

The individual ANN with LM algorithm and SVM with Polynomial kernel function were compared to know their performance. SVM showed better performance compared to ANN but a room for improvement was available. Hence, in continuation hybrid models were considered for the study. The different membership functions of the ANFIS and also different Kernel functions of PSO-SVM were compared before selecting the optimum function. Guass membership function for ANFIS and polynomial kernel for PSO-SVM shows better results compared to other membership

functions and kernel. Comparing the different hybrid models, ANFIS model performed better than PSO-SVM hybrid model.

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 SUMMARY

Damage level prediction of non-reshaped berm breakwater by considering all the boundary conditions and extracting knowledge from large amount of experimental or in-situ data is extremely difficult. And also, the physical model study is too expensive and time consuming. Furthermore, the relevant available information is usually in the form of prior empirical knowledge and input-output data of physical model studies. Therefore, we need an approximate reasoning system capable of handling such imperfect information. In this context soft computing models are developed, such as, ANN, SVM, PSO-SVM and ANFIS.

The data for the development of soft computing models were collected from physical model study on non-reshaped berm breakwater carried out in wave flume by Balakrishna Rao (2009). Non-dimensional input parameters that influence the damage level (S) of non-reshaped berm breakwater, such as, deep-water wave steepness (Ho/Lo), surf similarity parameter (ξ), relative berm position (h_B/d), relative armor stone weight (W_{50}/W_{max50}), relative berm width (B/Lo) and relative berm location (h_B/Lo) were used as the inputs for the soft computing techniques.

Initially ANN1 model was developed with different algorithms. PC analysis was carried out to study the influence of individual input parameters on the output. After PC analysis ANN2 model was developed with six input parameters. Here, the number of hidden layers was optimized to get better results. Further, SVM models with different kernel functions were developed. The individual ANN and SVM were compared to know their performance. SVM showed better performance compared to ANN but a room for improvement was available. Hence, in continuation hybrid models namely ANFIS and PSO-SVM were developed. All the models were compared in terms of statistical measures and computational time. Based on the present results conclusions were drawn and presented here.

5.2 CONCLUSIONS

There is no mathematical model to determine the damage level of breakwaters. Therefore, it is necessary for researchers to adopt the physical model study to quantitatively determine the parameters that influence the phenomenon. However, there are some difficulties in physical modeling that makes it inconvenient to use. Physical modeling is a time consuming process and expensive. Therefore different soft computing models namely, ANN, SVM, PSO-SVM and ANFIS were developed. And the results are compared in terms of CC, RMSE, SI and computational time. Based on the results of the present investigations and discussion thereon, following conclusions are arrived at:

- Based on the experimental data obtained by Balakrishna Rao(2009), seven input parameter namely relative armor stone weight, slope, wave steepness, surf similarity parameter, relative water depth, relative berm width and relative berm location was used to develop ANN1 model.

- From PCA study, it is observed that the slope is the least influencing parameter and after eliminating the slope, with the remaining six input parameters ANN2 model was developed. From the results variation of Correlations Coefficients (CC) between ANN1 and ANN2 is negligible.

- Soft computing models show good results in terms of statistical measures like Root Mean Square Error (RMSE), Correlation Coefficient (CC) and scatter index for observed and predicted damage level.

- A high CC is obtained at epoch equal to 300 with hidden nodes equal to five for ANN1 model whereas, ANN2 model has shown the similar trend with hidden nodes equal to five. In both models, non-linear transfer function tansig and linear transfer function purelin are used.

- SVM with different kernel function were studied and among them polynomial kernel gives higher CC and lower RMSE compared to other kernel functions.

- It is observed that SVM solution is unique. If we run the same program, the results will be exactly same for all SVM models whereas it is not possible in ANN.

- It is observed that hybrid models, such as PSO-SVM and ANFIS perform better than ANN and SVM.

- The performance of PSO-SVM appears to be highly influenced by the choice of its kernel function, and the good setting of kernel and SVM parameters. The polynomial kernel function performed superior than other kernel functions. It is also concluded that the parameter selection in the case of PSO-SVM has a significant effect on the performance of the model.

- The efficiency of the ANFIS models depends on the number of membership functions associated with each input data. Highest correlation coefficient was obtained using the two gauss shaped membership function.

- The hybrid models in both cases (ANFIS and PSO-SVM) showed better results compared to individual (ANN and SVM) models. When the hybrid models are compared, ANFIS model gives higher CC and lower RMSE. But considering computational time ANFIS has taken more time than PSO-SVM model. Hence PSO-SVM is computationally efficient as compared to ANFIS.

- An ANFIS and PSO-SVM model performs better and similar to observed values. Hence, ANFIS or PSO-SVM can replace the ANN, SVM for damage level prediction of non-reshaped berm breakwater.

- ANFIS or PSO-SVM can be utilized to provide a fast and reliable solution in prediction of the damage level prediction of non-reshaped berm breakwater, thereby making ANFIS or PSO-SVM as an alternate approach to map the wave structure interactions of berm breakwater.

5.3 SUGGESTIONS FOR FUTURE WORK

There is a scope for carrying out further research. The following suggestions may be considered for further study:

- Damage level of non-reshaped berm breakwater with random waves may be studied.
- Damage level of non-reshaped berm breakwater with different armors may be studied with random waves.
- Soft computing techniques may also be applied to find recession on reshaped berm breakwater.
- Latest soft computing techniques, such as, ant colony optimization, kernel principal component analysis, artificial immune system etc., may be carried out for predicting the damage level of non-reshaped berm breakwater.

APPENDIX

APPENDIX A: PROGRAM TO DEVELOP ANN MODEL

```
%Loading Input Data

Load D:\NITK\data\ Input\p.xls

Load D:\NITK\data\\Target\t.xls

% Selecting Input and Output Data for Training, Validation and Testing

[R,Q] = size (p);

iitst = 4:4:Q;

iival = 0:0:Q;

iitr = [1:4:Q 2:4:Q 3:4:Q];

% Dividing Input and Output Data for Training

ptr = p(:,iitr);

ttr = t(:,iitr);

% Dividing Input and Output Data for Validation

validation.P = p (:,iival);

validation.T = t (:,iival);

% Dividing Input and Output Data for Testing

testing.P = p (:,iitst);

testing.T = t (:,iitst);

% Creating a Feedforward Network
```

```
net = newff(ptr,ttr,[5 1],{'tansig'  'purelin'},'trainlm');
%Network Parameters
net.trainParam.show=5;
net.trainParam.epochs=300;
net.trainParam.min_grad =1e-100;
net.trainParam.goal=1e-5;
%Training the Network
[net,tr]=train (net,ptr,ttr,[],[],validation,testing);
%Simulate the Network with Train Data.
Outputs = sim(net,ptr);
%Simulate the Network with Test Data.
y= sim(net,testing.P);
%Calculation of Training Error
e=ttr-Outputs
perf = mse(e,Outputs,net)
%Calculation of Testing Error
e1=testing.T-y
perf = mse(e1,y,net)
% Plot the Correlation Coefficient for Training
plotregression(ttr,Outputs);
```

```
%Label X and Y axis

xlabel('Training Observed Value');

ylabel(‘TrainingPredicted Value');

%Plot the Correlation Coefficient for Testing.

plotregression(y,testing.T);

%Label X and Y axis

xlabel('Testing Observed Value');

ylabel(‘Testing Predicted Value');

%plots the training and test performances given the training record tr returned by the function train.

plotperform(tr)

% plots the training state from a training record tr returned by train.

plottrainstate(tr)

% To obtain weights between input and hidden layer

net.IW{1,1}

% To obtain weights between hidden and output layer

net.LW{2,1}

% To obtain bias between input and hidden layer

net.b{1,1}

%  To obtain bias between hidden and output layer

Net.b{2,1}
```

APPENDIX B: PROGRAM TO DEVELOP SVM MODEL

```
%Loading Input Data

Load D:\NITK\data\ Input\p.xls

Load D:\NITK\data\\Target\t.xls

%Selecting Input and Output Data for Training, Validation and Testing.

[R,Q] = size(p);

iitst = 4:4:Q;

iival = 0:0:Q;

iitr = [1:4:Q 2:4:Q 3:4:Q];

%Dividing Input and Output Data for Training.

ptr = p(:,iitr);

ttr = t(:,iitr);

%Dividing Input and Output Data for Validation.

validation.P = p(:,iival);

validation.T = t(:,iival);

%Dividing Input and Output Data for Testing.

testing.P = p(:,iitst);

testing.T = t(:,iitst);

% Initially assign the kernel function that is used, we have experimented with six
% kernel functions

ker='poly';
```

```
% Here the kernel parameters are assigned

global p1 p2;

p1=3;

%SVM  Support Vector Machine, here 15 is capacity factor, and 0.001 is error tube

% Performance of SVM depends on good setting of SVM and Kernel parameters

[nsv, beta, bias] = svr (ptr, ttr, ker, 15,'quadratic', 0.001);

% SVM model give the predicted test output

ptestoutput = svroutput (testing.P, testing.T, ker, beta, bias);

% SVM model give the predicted train output

ptrainoutput = svroutput (ptr, ptr, ker, beta,bias);

%Plot the Correlation Coefficient for Training.

plotregression(ttr, ptrainoutput);

%Label X and Y axis

xlabel('Training Observed Value');

ylabel(‘Training Predicted Value');

%Plot the Correlation Coefficient for Testing.

plotregression (ptestoutput, testing.T);

%Label X and Y axis.

xlabel('Testing Observed Value');

ylabel(‘Testing Predicted Value');
```

APPENDIX C: MATLAB PROGRAMME FOR PSO-SVM

```
% Assign number of generation and population size

gen_no=’no. of generation’;

Np=’no. of population size’;

%Initialization

pos=rand(Np,1);

vel=rand(Np,1)-0.5;

% Generation loop

for t=1:gen_no

   init_pop=[pos,vel];

   pop=init_pop;

   for i=1:Np

      ker='poly';   global p1;

      p1=3.9;

      d1=pop(i,1);   e1=pop(i,2);

%SVM Support Vector Machine

      [nsv, beta, bias] = svr(X,Y,ker,d1,'quadratic',e1);

      ptrainoutput = svroutput(X,X,ker,beta,bias);

      cctrain=corrcoef(ptrainoutput,Y);

      ptestoutput = svroutput(X,X1,ker,beta,bias);
```

```
        cctest=corrcoef(ptestoutput,Y1);

        MSE_Pos(i)=mse(Y1-ptestoutput);
    end
    end
    %Best solutions (pbest)
    Pbest=(MSE_Pos)';
    [min_fit_Pbest Index] = min(Pbest);
%gbest
Gbest=Pbest(Index,:) ;
%updating position and velocity
    iw = 0.9 - (0.9-0.4)*t/ gen_no;
    for j=1:Np
        vel(j,:)=iw*vel(j,:)+0.2*rand*(Pbest(j,:)-pos(j,:))+2.5*rand*(Gbest-pos(j,:));
        pos(j,:)=pos(j,:)+vel(j,:);
%Fitness Evaluation
        if (Pbest(j)>pos(j))
            Pbest(j,:)=pos(j,:);
        end
        if (Gbest>Pbest(j))
            Gbest=Pbest(j,:);
        end
    end
end
```

APPENDIX D: MATLAB PROGRAMME FOR ANFIS

```
%Define the No of Membership Functions.

number_mf=2;

%Define the Type of Membership Function.

mf_type='gaussmf';

%Generating the initial Fuzzy Inference System.

initial_fis = genfis1(train_data, number_mf, mf_type)

%Plots  of Initial Membership Functions that are equally spaced and cover the whole
input space .

plotmf(initial_fis, 'input', 1)

plotmf(initial_fis, 'input', 2)

plotmf(initial_fis, 'input', 3)

plotmf(initial_fis, 'input', 3)

plotmf(initial_fis, 'input', 4)

plotmf(initial_fis, 'input', 5)

plotmf(initial_fis, 'input', 6)

%Define the No of Iterations.

number_iterations=100;

%Train the Network.

[train_out_fis train_error step_size test_fis_out test_error] = ...

anfis(train_data, initial_fis, number_iterations, [1,1,1,1], test_data)

%Obtain the MinimumTraining and Minimum Testing Error Value.
```

```
train_error1=min(train_error)

test_error1=min(test_error)

%Plot the Training and Testing Error.

plot([train_error test_error]);

hold on; plot([train_error test_error], 'o'); hold off;

%Label X and Y axis.

xlabel('Epochs');

ylabel('RMSE (Root Mean Squared Error)');

%Title and Legend for the plot.

title('Error Curves');

legend('ANFIS Training output','ANFIS Testing Output')

%To evaluate the Training output of a ANFIS system for a given input.

Output.TR=evalfis(ptr,train_out_fis);

%Plot Correlation Coefficient for Training.

plotregression(ttr,Output.TR);

%Label X and Y axis.

xlabel('Training Target')

ylabel('ANFIS Training ouput')

%To evaluate the Training output of a ANFIS system for a given input.

Output.Tst=evalfis(testing.P,test_fis_out);
```

```
%Plot Correlation Coefficient for Testing.

plotregression(testing.T,Output.Tst);

%Label X and Y axis.

xlabel('Testing Target')

ylabel('ANFIS Testing ouput')
```

REFERENCES

Ahmadi, M. A., Shadizadeh, S. R. and Hasanvand, M. (2011). "Neural network based stochastic particle swarm optimization for prediction of minimum miscible pressure". *International Journal of Computer Applications*, 34(1), 15-19.

Ahrens, J. P. (1970). "The influence of breakwater type on rip rap stability."*Proc., 12th ICCE*, Washington, 3, 1557 – 1566.

Ahrens, J. P. (1984). "Reef type breakwaters."*Proc. 19th ICCE*, Houston, USA, 2648 – 2662.

Andersen, L. T. and Burcharth, H. F. (2010). "A new formula for front slope recession of berm breakwaters." Coastal Engineering, 57(4), 359–374.

Aydogan, B., Ayat, B., Ozturk, M. N., Cevik E. O. and Yuksel, Y. (2010). "Current velocity forecasting in straits with artificial neural networks, a case study: Strait of Istanbul." *Ocean Engineering*, 37, 443 – 453.

Baird, W. F. and Hall, K. R. (1984). "The Design of breakwaters using quarried stones." *Proc., 19th Int. Conf.on Coastal Engineering*, 2580-2591.

Balas, C. E., Koç, M. L. and Tür, R. (2010). "Artificial neural networks based on principal component analysis, fuzzy systems and fuzzy neural networks for preliminary design of rubble mound breakwaters". *Applied Ocean Research*, 32(4), 425-433.

Balakrishna Rao, (2009). "Experimental investigation on stability of statically stable berm breakwater".Doctorial thesis, National Institute of Technology Karnataka, Surathkal, Mangalore, India.

Bateni, S. M. and Jeng, D. S. (2007). "Estimation of pile group scour using adaptive neuro-fuzzy approach." *Ocean Engineering*, 34, 1344-1354.

Beltrami, G. M.(2008). "An ANN algorithm for automatic, real-time tsunami detection in deep-sea level measurement."*Ocean Engineering*, 35, 572-587.

Browne, M., Castelle, B., Strauss, D., Tomlinson, R., Blumenstein, M. and Lane, C. (2007). "Near-shore swell estimation from a global wind-wave model: spectral process, linear and artificial neural network." *Coastal Engineering*, 54, 445-460.

Bruun, P. and Gunbak, A. R. (1978). "Stability of sloping structures in relation to $\xi = \tan\alpha/\sqrt{(H/Lo)}$ risk criteria in design." *J. Coastal Engineering*, 1, 287 – 322.

Carver, R. D. and Davidson, D. D. (1982). "Breakwater stability – Breaking wave data."*Proc., Coastal Engg. Conf.,* 2107 – 2128.

Chang, H. K. and Chien, W. A. (2006)."A fuzzy neural hybrid system of simulating typhoon waves."*Coastal Engineering*, 53, 737-748.

Chang, H. K. and Lin, L. C. (2006)."Multi-point tidal prediction using artificial neural network with tide-generating forces."*Coastal Engineering*, 53, 857-864.

Chau, K. W. (2006). "Particle swarm optimization training algorithm for ANNs in stage prediction of ShingMun River". *Journal of hydrology*, 329(3), 363-367.

Chen, Q., Chen, X. and Wu, Y. (2010)."Optimization Algorithm with Kernel PCA to Support Vector Machines for Time Series Prediction". *Journal of Computers,* 5(3), 380-387.

Coastal Engineering Manual.(CEM).(2006). "Fundamental of Design."EM 1110-2-1100(Part-VI), U.S. Army Corps.of Engineers, Vicksburg, Miss.

Das, N., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M. andBasu, D. K. (2012)."A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application". *Applied Soft Computing*, *12*(5), 1592-1606.

Deo, M. C. and Kumar, K. N. (2000)."Interpolation of wave heights."*Ocean Engineering*, 27, 907-919.

Deo, M. C. and Jagdale, S. S. (2003). "Prediction of breaking waves with neural networks." *Ocean Engineering*, 30, 1163-1178.

Deo, M. C., Jha, A., Chaphekar A. S. and Ravikant, K. (2001). "Neural networks for wave forecasting." *Ocean Engineering*, 28, 889-898.

Ergin, A. and Pora, S. (1971). "Irregular wave action on rubble mound breakwaters." *J. Waterways, Port, Coastal, Ocean Eng*., ASCE, 97(WW2), 279-293.

Ergin, A., Gunbak, A. R. and Yanmaz, A. M. (1989). "Rubble mound breakwaters with S-shape design." *J. Waterway, Port, Coastal, Ocean Eng*., ASCE, 115(5), 579-593.

Gadre, M. R., Somayaji, N. and Kale, A. G. (1985). "Stones in breakwaters and seawalls."*Proc., 1st Nat.Conf.on Dock and HarbourEngg.,* IIT Bombay, Dec., 1, B-107 – B-112.

Gaur, S. and Deo, M. C. (2008)."Real-time wave forecasting using genetic programming."*Ocean Engineering*, 35, 1166-1172.

Gholami, R., Shahraki, A. R. and Paghaleh, J. M. (2012)."Prediction of Hydrocarbon Reservoirs Permeability Using Support Vector Machine".*Mathematical Problems in Engineering*, 2012, 1-18.

Gunaydin, K. (2008). "The estimation of monthly mean significant wave heights by using artificial neural network and regression methods."*Ocean Engineering*, 35, 1406-1415.

Gunn,S. (1997)."Support Vector Machines for Classification and Regression", ISIS Technical Report.

Guven, A., Azamathulla, H. Md. and Zakaria, N. A. (2009). "Linear genetic programming for prediction of circular pile scour." *Ocean Engineering*, 36, 985-991.

Hagras, M. A. (2013). "Prediction of hydrodynamic coefficients of permeable paneled breakwater using artificial neural networks". *International Journal of Engineering Science & Technology*, 5(8).

Hall, K. R.. and Kao, J. S. (1991). "The influence of armor stone gradation on dynamically stable breakwaters."*J. Coastal Engineering*, 15, 333-346.

Hagan, M. and Menhaj, M. (1994). "Training feedforward networks with the Marquardt algorithm." *IEEE Transactions on Neural Networks*, 5, pp. 989–993.

Hannoura, A. A. A. and McCorquodale, J. A. (1985). Rubble mounds: numerical modeling of wave motion. Journal of Waterway, Port, Coastal, and Ocean Engineering, 111(5), 800-816.

Hashemi M. R., Ghadampour Z. and Neill S. P. (2010)."Using an artificial neural network to model seasonal changes in beach profiles."*Ocean Engineering*, 37, 1345 – 1356.

Hegde, A. V. and Samaga, B. R. (1996). "Study on the effect of core porosity on rubble mound breakwater." *Proc., 10th Congress of the Asian and Pacific Regional Division of the Int. AssnFor Hydraulic Research*,Langkawi, Malaysia, 1, 287 – 294.

Huang, W., Murray, C., Kraus, N. and Rosati, J. (2003). "Development of a regional neural network for coastal water level predictions."*Ocean Engineering*, 30, 2275-2295.

Hudson, R. Y. (1959). "Laboratory investigation of rubble mound breakwaters." *J. waterway and Harbour Division*, 85(3), 93 – 121.

Iglesias, G., Castro, A. and Fraguela, J. A. (2010). "Artificial intelligence applied to floating boom behaviour under waves and currents". *Ocean Engineering*, 37, 1513 – 1521.

Jang J. S. R., (1993). "ANFIS: Adaptive Network Fuzzy Inference System". *IEEE Transactions on Systems, Man, and Cybernetics* 23 (3), 665-685.

Jian-sheng, W. U. and Long, J. I. N. (2009). "Study on the Meteorological Prediction Model Using the Learning Algorithm of Neural Ensemble Based on PSO Algorithms". *Journal of tropical meteorology13*(1), 83-88.

Johnson, R. R., Mansard, E. P. D. and Ploeg, J. (1978)."Effect of wave grouping on breakwater stability."*Proc. 10th Int. Conf. on Coast.Eng.*, ASCE, 2228-2247.

Kalra, R. and Deo, M. C. (2007)."Derivation of coastal wind and wave parameters from offshore measurements of TOPEX satellite using ANN."*Coastal Engineering*, 54, 187-196.

Karla, R., Deo, M. C., Kumar, R. and Agarwal, V. K. (2005)."RBF network for spatial mapping of wave heights."*Marine Structures*, 18, 289-300.

Kazeminezhad, M. H., Etemad-shahidi, A. and Mousvi, S. J. (2005)."Application of fuzzy inference system in the prediction of wave parameters."*Ocean Engineering*, 32, 1709-1725.

Kennedy J. and Eberhart, R. (1995)."Particle Swarm Optimization", *Proc. IEEE Int. Conf. on Neural Networks*, Perth, 1492-1948.

Kecman, V. (2001). "Learning and Soft Computing, Support Vector Machines", Neural Networks, and Fuzzy Logic Models, The MIT Press, Cambridge, MA, USA.

Kim, D. H. and Park, W. S. (2005). "Neural network for design and reliability analysis of rubble mound breakwaters." *Ocean Engineering*, 32, 1332-1349.

Kim, D., Kim, D. H., Chang, S.and Lee, J. J. (2011). "Stability number prediction for breakwater armor blocks using Support Vector Regression". *KSCE Journal of Civil Engineering*, 15(2), 225-230.

Kobayashi, N. and Jacobs, B. (1985). "Riprap stability under wave action." *J. Waterway, Port, Coastal, Ocean Eng.*, ASCE, 111 (3), 552-566.

Kobayashi, N., Farhadzadeh, A. and Melby, J. A. (2010). "Wave overtopping and damage progression of stone armor layer". *Journal of Waterway, Port, Coastal, and Ocean Engineering*, *136*(5), 257-265.

Koç, M. L. and Balas, C. E. (2013). "Reliability analysis of a rubble mound breakwater using the theory of fuzzy random variables".*Applied Ocean Research*, 39, 83-88.

Kondo, H., Toma, S. and Yanno, K. (1976). "Laboratory study on pervious core breakwaters."*Proc., 15th ICCE,* 3, 2643 – 2661.

Lee, T. L. and Jeng, D. S. (2002)."Application of artificial neural networks in tide forecasting."*Ocean Engineering*, 29, 1003-1022.

Lee, T. L. (2004). "Back propagation neural network for long term tidal predictions."*Ocean Engineering*, 31, 225-238.

Liang, S., Li, M. and Sun, Z. (2008). "Prediction models for tidal level including strong meteorologic effects using neural network." *Ocean Engineering*, 35, 666-675.

Londhe, S. N. (2008). "Soft computing approach for real-time estimation of missing wave heights."*Ocean Engineering*, 35, 1080-1089.

Mahjoobi, J. and Mosabbeb E. A. (2009)."Prediction of significant wave height using regressive support vector machines."*Ocean Engineering*, 36, 339-347.

Makarynskyy, O. (2004). "Improving wave predictions with artificial neural networks."*Ocean Engineering*, 31, 709-724.

Makarynskyy, O., Pires-Silva, A.A., Makarynska, D.and Ventura-Soares, C. (2005)."Artificial neural networks in wave predictions at the west coast of Portugal". *Computers & Geosciences*, 31(4), 415-424.

Mandal, S., Rao, S. and Manjunatha, Y.R. (2007). "Stability analysis of rubble mound breakwater using ANN." *Proc., Indian National Conference on Harbour and Ocean Engineering, INCHOE*, National Institute of Technology Karnataka, Surathkal, India, 551-560.

Mandal, S., Rao, S. and Prabhaharan, N. (2001)."Wave forecasting using neural networks."*Proc., International Conference in Ocean Engineering, ICOE*, Indian Institute of Technology, Madras, India, 103-108.

Mandal, S. and Prabaharan, N. (2006). "Ocean wave forecasting using recurrent neural network." *Ocean Engineering*, 33, 1401-1410.

Mandal, S., Rao, S. and Raju, D.H. (2005). "Ocean wave parameters estimation using back propagation neural network." *Marine Structures*, 18, 301-318.

Mandal, S., Rao, S. and Manjunatha, Y. R. (2008)."Stability prediction of berm breakwater using neural network."*Proc., International conference on COPEDEC VII*, Dubai (UAE), 27, 1-11.

Mase, H., Sakamoto, M. and Sakai, T. (1995). "Neural network stability of rubble mound breakwaters." *Journal of Waterway, Port, Coastal and Ocean Engineering*, ASCE, 121(6), 294-299.

Mase, H. and Kitano, T. (1999) "Prediction model for occurrence of wave force." *Ocean Engineering*,26**,** 949–961

Masters, T. (1995). "Advanced Algorithms for Neural Networks: C++ Sourcebook." John Wiley, Frankfurt, Germany.

Math Works, MATLAB – The Language of Technical Computing, URL http://www.mathworks.com/, 2011.

Melby, J. A. and Kobayashi, N. (1998)."Progression and variability of damage on rubble mound breakwaters". *J. of waterway, port, coastal, and ocean engineering*, 124(6), 286-294.

Melby, J. A. and Hughes, S. A. (2004). "Armor stability based on wave momentum flux." *Proc., Int.. Conf. on Coastal Structures,* Portland, Oregan, Aug. 53-65

Mirchandani, G., Cao, W. and Bosworth, B. (1989)." Efficient implementation of neural nets using an optimal relationshipbetween number of patterns, input

dimension and hidden nodes". *In Acoustics, speech abd signal processing, ICASSP-89, Inter. Conference* IEEE, 2521-2523.

Møller, M. F. (1993). "Supervised Learning On Large Redundant Training Sets".*J. Neural System* 4(1), 15-25.

Naithani, R. and Deo, M. C. (2005). "Estimation of wave spectral shapes using ANN." *Advances in Engineering Software*, 36, 750-756.

Pan, M., Zeng, D. and Xu, G. (2010)."Temperature prediction of hydrogen producing reactor using SVM regression with PSO".*Journal of computers*, 5(3), 388-393.

PIANC (2003)."State-of-the-art of the design and construction of berm breakwaters."*PIANC, Rep. of Working Group 40 – MARCOM,*Brussels, Belgium.

Poonawala, I. Z., Kudale, M. D. and Purohit, A. A. (1994). "Effect of seabed slope on stability of armor layer."*Proc., Indian Nat. Conf. of Harbour and Ocean Eng.*, 2, B-97 – B-104.

Priest, M. S., Pugh, J. W. and Singh, R. (1964). "Seaward profile for rubble mound breakwaters." *Proc., 9th Int. Conf. on Coastal Engineering,* ASCE, 553-559.

Rajasekaran, S., Jeng, D. S. and Lee, T. L. (2005)."Tidal level forecasting during typhoon surge using functional and sequential learning neural networks."*Journal of Waterway, Port, Coastal and Ocean Engineering*, ASCE, 131(6), 321-324.

Rajasekaran, S., Thiruvenkataswamy, K. and Lee, T. L. (2006)."Tidal level forecasting using functional and sequential learning neural networks."*Applied Mathematical Modeling*, 30, 85-103.

Rao, S. and Mandal, S. (2005). "Hind casting of storm waves using neural networks." *Ocean Engineering*, 32, 667-684.

Rao, S., Pramod, C. and Rao, B. (2004)."Stability of berm breakwater with reduced armor stone weight". *Ocean engineering*, 31(11), 1577-1589.

Rao, S., Subrahmanya, K., Balakrishna Rao, K. and Chandramohan, V. R. (2008)."Stability aspects of non-reshaped berm breakwaters with reduced armor weight". *J. of waterway, port, coastal, and ocean engineering*, 134(2), 81-87.

Rao, S. (2000)."Studies on performance of perforated pile breakwater."Doctoral dissertation, Mangalore University, Mangalore, India.

Shi, X. C. and Dong, Y. F. (2011).“Support vector machine applied to prediction strength of cement”. *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2nd International Conference,* IEEE.1585-1588.

Sigurdarson, S., Viggosson, G., Einarsson, S. and Smarason, O. B. (1998).“Berm Breakwaters, fifteen years experience.”*Proc. 25th Int. Con. on Coastal Engineering,* Copenhagen, Denmark, ASCE, 36 – 37.

Smola, A.J. and Schölkopf, B. (1998)."A Tutorial on Support Vector Regression". *NeuroCOLT Technical Report* NC-TR-98-030, Royal Holloway College, University of London, UK.

Shore Protection Manual.(SPM).(1977). U.S. Army Corps of Engineers, CERC, Washington, D.C.

Shore Protection Manual.(SPM).(1984).U.S. Army Corps of Engineers, CERC, Washington, D.C.

Sylaios, G., Bouchette, F., Tsihrintziz, V.A. and Denamiel C. (2009). “A fuzzy inference system for wind wave modeling.” *Ocean Engineering*, 36, 1358-1365.

Takahashi, S. (1996), “Design of Vertical Breakwaters”, Reference Document no.34, Port and Harbour Research Institute, Ministry of Transport, Yokosuka, Japan

Timco, G. W., Mansard, E. P. D. and Ploeg, J. (1984).“Stability of breakwaters with variations in core permeability.”*Coastal Engineering,* 2487-2499.

Torum, A. (1998). “On the stability of berm breakwaters in shallow waters and deep water.”*Proc., 26th Int. Conf. on Coastal Engineering,* Copenhagen, Denmark, ASCE, 1435-1448

Torum, A., Krogh, S. R., Bjordal, S., Fjeld, S., Archetti, R. and Jacobsen, A. (1999). ”Design criteria and design procedure for berm breakwaters.” *Proc., Coastal Structures ‘99’,*Balkema, Rotterdam, 331-341.

Torum, A., Franziska, K. and Menze, A. (2003).“On berm breakwaters.stability, scour, overtopping.” *J. Coastal Engineering,* 49, 209-238.

Tripathy, A., Mohapatra, S., Beura, S. and Pradhan, G. (2011).“Weather Forecasting using ANN and PSO”.*Int. J. Sci. Eng. Res, 2,* 1-5.

Van Gent, M. R. A. (1995). "Wave interaction with berm breakwaters."*J. Waterway, Port, Coastal, Ocean Eng.*, ASCE, 121 (5), 229-238.

Van Gent, M. R. (2013). "Rock stability of rubble mound breakwaters with a berm".*Coastal Engineering*, 78, 35-45.

Van Gent, M. R., van den Boogaard, H. F., Pozueta, B. and Medina, J. R. (2007). "Neural network modelling of wave overtopping at coastal structures".*Coastal Engineering*, 54(8), 586-593.

Van der Meer, J. W. (1988). "Rock slopes and gravel beaches under random wave attack." Doctorial thesis, Delft University of Technology, The Netherlands.

Van der Meer, J. W. and Pilarczyk, K. W. (1984). "Stability of rubble mound slops under random wave attack." *Proc., 19th ICCE,* Houston, 2620-2634.

Van der Meer, J. W. (1992). "Stability of the seaward slope of berm breakwaters."*J. Coastal Engineering*, 16, 205-234.

Van der Meer, J. W., Angremond, K., and Gerding, E. (1996). "Toe structure stability of rubble mound breakwaters." Advances in Coastal Structures and Breakwaters, Thomas Telford Ltd., London, 308 – 321.

Vapnik, V. (1995)."The Nature of Statistical Learning Theory", *Springer Verlag*, New York.

Vapnik,V., Golowich, S. and Smola, A.(1996)."Support vector method for function approximation regression estimation and signal processing", *Adv. in Neural Inform.Proces.Syst*, 9 281-287.

Vairappan, C., Tamura, H., Gao, S. and Tan Z., (2009). "Batch type local search-based adaptive neuro-fuzzy inference Systems (ANFIS) with self-feedbacks for time series prediction". *J. Neuro computing* 72, 1870-1877.

Xi, Z., Zhang, Y. and Zhu, C. (2012)."Application of PSO-Neural Network Model in Prediction of Groundwater Level in Handan City".*Advances in Information Sciences & Service Sciences*, 4(6).177-185.

Yagci, O., Mercan, D. E., Cigizoglu and Kabdasli, M. S. (2005). "Artificial intelligence methods in breakwater damage ratio estimation." *Ocean Engineering*, 32, 2088-2106.

Yoon, H., Jun, S-C., Hyun, Y., Bae, G-O. and Lee, K-K. (2011). “A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer”. *Journal Hydrology*, 396, 128–138.

Zamani, A., Solomatine, D., Azimian, A. and Heemink, A. (2008).“Learning from data for wind-wave forecasting.”*Ocean Engineering*, 35, 953-962.

Zanuttigh, B., Formentin, S. M., and Briganti, R. (2013).“A neural network for the prediction of wave reflection from coastal and harbor structures”. *Coastal Engineering*, 80, 49-67.