

# **Modeling and Analysis of Ready Mix Concrete Slump using Hybrid Genetic Algorithms-Artificial Neural Networks**

**Ph.D. Thesis**

**Vinay Chandwani**

**ID: 2011RST7139**



Department of Civil Engineering

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JAIPUR-302017

January, 2017

# **Modeling and Analysis of Ready Mix Concrete Slump using Hybrid Genetic Algorithms-Artificial Neural Networks**

**This thesis is submitted  
as a partial fulfillment of the Ph.D. programme in Engineering**

**Vinay Chandwani  
ID: 2011RST7139**



Department of Civil Engineering  
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR  
JAIPUR-302017  
January, 2017

**© Malaviya National Institute of Technology Jaipur  
All rights reserved**



MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

## Certificate

This is to certify that the thesis report entitled “**Modeling and Analysis of Ready Mix Concrete Slump using Hybrid Genetic Algorithms-Artificial Neural Networks**” which is being submitted by **Vinay Chandwani, ID: 2011RST7139**, for the partial fulfillment of the degree of **Doctor of Philosophy** in Civil Engineering to the Malaviya National Institute of Technology Jaipur has been carried out by him under my supervision and guidance. I consider it worthy of consideration for the award of the degree of Doctor of Philosophy of the institute.

**(Dr. Vinay Agrawal)**

Supervisor

**(Prof. Ravindra Nagar)**

Supervisor

The Ph.D. viva voce examination of Mr. Vinay Chandwani has been conducted by the Oral Defense Committee (ODC) constituted by the Chairman SPGB, as per 9.4.3, vide letter No. F.4(P) PhD /Acad./MNIT/2016/1355 dated 07.12.2016. The ODC declares that the student has successfully defended the thesis in the viva-voce examination.

**(Dr. Vinay Agrawal)**

Supervisor

**(Prof. Ravindra Nagar)**

Supervisor

**(Dr. Rajan Choudhary)**

External Examiner

Date: \_\_\_\_\_



MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

### **Candidate's declaration**

I hereby certify that the work which is being presented in the thesis entitled '**Modeling and Analysis of Ready Mix Concrete Slump using Hybrid Genetic Algorithms-Artificial Neural Networks**' in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy and submitted to the Department of Civil Engineering, Malaviya National Institute of Technology Jaipur, is an authentic record of my own work carried out at Department of Civil Engineering during a period from December 27, 2011 to April 15, 2016 under the supervision of Dr. Vinay Agrawal, Assistant Professor, Department of Civil Engineering and Dr. Ravindra Nagar, Professor, Department of Civil Engineering, Malaviya National Institute of Technology Jaipur.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Date: \_\_\_\_\_

**(Vinay Chandwani)**

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

**(Dr. Vinay Agrawal)**  
**(Supervisor)**

Assistant Professor,  
Department of Civil Engineering,  
Malaviya National Institute of Technology Jaipur.  
Jaipur- 302017, Rajasthan, India.

Date: \_\_\_\_\_

**(Dr. Ravindra Nagar)**  
**(Supervisor)**

Professor,  
Department of Civil Engineering,  
Malaviya National Institute of Technology Jaipur.  
Jaipur- 302017, Rajasthan, India.

Date: \_\_\_\_\_

## **Acknowledgement**

I humbly grab this opportunity to acknowledge reverentially, may people who deserve special mentions for their varied contributions in assorted ways that helped me during my Ph.D. research and the making of this thesis. I could never have embarked and finished the same without their kind support and encouragements.

First and foremost, I would like to express my sincere gratitude and praise to the Almighty GOD, who had showered his grace in the form of knowledge and wisdom and every other way for completing this thesis.

I would like to express my profound gratitude to my guides Dr. Vinay Agrawal, Assistant Professor, Department of Civil Engineering and Dr. Ravindra Nagar, Professor, Department of Civil Engineering, Malaviya National Institute of Technology, Jaipur, Rajasthan, India for their supervision, advice, and invaluable guidance from the very early stages of this research. Their perceptual inspiration, encouragement, and understanding have been a mainstay of this work. From their busy schedule, they always spared time for assessing the progress of my work. Their wide knowledge regarding the subject helped me in writing this thesis. I am indebted for their kind help and support which made it possible for me to stand up to the challenges offered by the task and come out successfully.

I am thankful to the Officers and Staff of academic affairs for their cooperation in academic work and help throughout the course of study. I am not able to find words in any of the dictionaries for thanking Prof. Gunwant Sharma, Prof. Y.P. Mathur, Prof. A.B. Gupta, Prof. A.K.Vyas, Prof. Sudhir Kumar and Dr. Sandeep Choudhary, Department of Civil Engineering, MNIT Jaipur for their valuable guidance, unflinching encouragement, keeping my moral high during the course of the work and helped me out whenever I needed them. I am extremely thankful to members of DGPC and DREC, for their support and guidelines regarding my thesis work. I am also thankful to Dr. Mahesh Jat, Dr. S.K Tiwari, Dr. Pawan Kalla, Dr. Arun Gaur, Prof. Rohit Goyal, Prof. B.L. Swami, Dr. M. K. Shrimali, Dr. S.D.Bharti, Dr. J.K. Jain, Dr. Rajesh Gupta, Dr. Sanjay Bhattar and all other faculty members of Malaviya National Institute of Technology Jaipur.

I extend my deep sense of gratitude to Director, Head, Civil Engineering Department, Dean Academics, and Dean R&D of MNIT Jaipur for strengthening the research

environment of the Institute and for providing me with the environment and space to carry out my research work. I give my thanks to Mr. Rajesh Saxena, Office-in-Charge, Civil Engineering Department, MNIT Jaipur, Mr. Ramjilal Meena and Mr. Sher Singh, who were always ready to extend me every possible help throughout my work.

I deeply acknowledge Er. Raja Ram Yadav, Chief Engineer (Retd.), Er. Girish Lodha, Additional Chief Engineer, Water Resource Er. Naveen Kumar Gupta, Executive Engineer, Er. Sunil Vyas, Executive Engineer, Er. Kiran Ahuja, Executive Engineer; and all other engineers and staff of Water Resources Department, Government of Rajasthan, India and Er. Sanjay Mundra, NCCBM whose regular support, guidance, motivation and contribution towards my thesis cannot be unseen.

From the bottom of my heart, I ever realize the inspiration given by my father, Prof. D.T. Chandwani, and mother, Late Smt. Vidya Chandwani and my grandfather Late Sh. Tahilram Chandwani. I cannot forget to acknowledge here the tolerance and patience of the kids, Karan and Bhavika, which they kept during the progress of my research work. Lastly but not the least, I would like to acknowledge the efforts made by my wife, Mrs. Pooja Chandwani, in providing me with the environment to carry on my thesis work at home with an ease.

It is not possible for me to pen down my thanks to all those who helped me directly or indirectly from time in completing this task. Each help is like a brick, which contributes in building a structure.

Date: \_\_\_\_\_

**(Vinay Chandwani)**

Student ID: 2011RST7139

## Abstract

Workability of concrete measured quantitatively as slump value is one of the most significant quality assurance parameters in the Ready Mix Concrete (RMC) industry. It ensures that RMC transported with long delivery times is still in the state that it could be easily placed, compacted and finished at the construction site. Modeling slump based on proportions of the design mix constituents is challenging and demands a non-algorithmic approach due to composite nature of concrete. Multilayer Feedforward Neural Networks (MFNN) trained using backpropagation (BP) algorithms have been a preferred choice for the researchers for modeling the material behavior of concrete. However, the BP algorithm's likelihood of entrapment at local minima and slow convergence rate are major drawbacks during its implementation.

The study aims at exploring the usefulness of hybridizing two distinct Soft Computing techniques namely, Genetic Algorithms (GA) and Artificial Neural Networks (ANN), for modeling the relationship between the proportions of the RMC design mix constituents namely, cement, pulverized fuel ash (PFA) or fly ash, sand, coarse aggregate (20mm), coarse aggregate (10mm), superplasticizer and water and their corresponding initial slump measured at the batching plant. The study further attempts to (a) compare the prediction accuracy and reliability of the neural network and regression models; (b) analyze and explore the material behaviour of concrete slump; (c) assess the effectiveness and applicability of the slump model for a different RMC batching plant; and (d) develop a decision support tool to estimate the slump value for the concrete design mix.

The data for the research were collected from two different RMC batching plants located in the same city. The data collected from the first RMC batching plant were utilized to build the model for the concrete slump. The hybrid Genetic Algorithms–Levenberg-Marquardt Backpropagation Neural Network (GA-LMBNN) model utilized the global exploration ability of GA for evolving the optimal set of initial weights and biases for the MFNN, which was subsequently fine-tuned using the fast converging Levenberg-Marquardt (LM) backpropagation algorithm. Apart from the hybrid GA-LMBNN model, the conventional LMBNN and regression models were developed for the concrete slump. The developed models for concrete slump were compared using six different statistical performance metrics. Response trace plots and



connection weights method were utilized to provide insight into the material behavior of concrete slump and to ascertain the relative importance of individual design mix constituents of concrete on the slump value respectively. The effectiveness and universal applicability of the slump model were assessed by, utilizing it to predict the slump for concrete design mix data collected from the other RMC batching plant. The knowledge extracted from RMC data in the form of a neural network model, is used to develop a decision support tool for estimating initial slump for the concrete design mix.

As a result of hybridization, the GA, and LM backpropagation algorithm is shown to complement each other, robustly covering up the individual drawbacks, rendering fast speed of convergence coupled with a consistent and improved learning and generalization to the neural network model. The hybrid GA-LMBNN model for the slump is shown to outperform the learning and generalization accuracy of the conventional LMBNN, first order regression and second order regression models. In comparison to the traditional regression models, the neural network models are shown to exhibit better prediction accuracy, demonstrating their potential for modeling highly complex and unstructured functional relationship presented by the concrete's design mix constituents and slump value.

The sensitivity analysis using connection weights method reveals that superplasticizer and PFA exhibit greater relative importance on the slump value of RMC. The response trace plots indicate that there exists a certain critical level of each constituent beyond which a significant change in the behavior of concrete slump is noticeable. The performance of the slump model for the other RMC batching plant proved that, that the concrete slump is sensitive to both physical and chemical properties of the design mix constituents, and therefore, any minor change in the properties of the individual constituents can significantly affect the prediction accuracy of the slump model. The decision support tool is shown to provide a fair assessment of slump value for given concrete design mix and further provides liberty to the technical personnel to explore various mix proportions for formulating concrete design mix of the desired workability, without performing multiple slump tests at the RMC batching plant.

## Contents

Certificate	i
Candidate's declaration	ii
Acknowledgements	iii
Abstract	v
Contents	vii
List of Figures	xii
List of Tables	xvi
List of Symbols and Abbreviations	xviii
<b>Chapter 1 Introduction</b>	<b>1-13</b>
1.1 Background	1
1.1.1 Ready Mix Concrete (RMC)	1
1.1.2 Workability of concrete	1
1.1.2.1 Definition and importance	1
1.1.2.2 Workability assessment: Slump test	3
1.1.2.3 Factors affecting workability	4
1.1.3 Soft Computing techniques	5
1.1.3.1 Artificial Neural Networks: A modeling tool	6
1.1.3.2 Genetic Algorithms: An optimization tool	7
1.2 Need for the present study	7
1.3 Objectives of the present study	10

	1.4	Thesis organization	11
	1.5	Summary	12
<b>Chapter 2</b>		<b>Artificial Neural Networks and Genetic Algorithms: An Overview</b>	<b>14-39</b>
	2.1	Introduction	14
	2.2	Artificial Neural Networks	14
	2.2.1	Historical background	15
	2.2.2	Basic functional unit: An artificial neuron	16
	2.2.3	Characteristics	18
	2.2.4	Backpropagation neural networks	23
	2.3	Genetic Algorithms	27
	2.3.1	Historical background	28
	2.3.2	Structure and working: A biological perspective	29
	2.3.3	Operators	31
	2.3.4	Parameters	37
	2.4	Summary	38
<b>Chapter 3</b>		<b>Literature Survey</b>	<b>40-74</b>
	3.1	Introduction	40
	3.2	Artificial Neural Networks applications	41

3.2.1	Applications in modeling various properties of concrete	41
3.2.2	Applications in modeling slump of concrete	49
3.3	Hybrid Genetic Algorithms-Artificial Neural Network applications	55
3.3.1	Applications in modeling material behavior of concrete	56
3.3.2	Civil Engineering applications	58
3.3.3	Multidisciplinary applications	72
3.4	Conclusions of literature survey	73
3.5	Research gap	74
<b>Chapter 4</b>	<b>Modeling and Analyzing Concrete Slump</b>	<b>75-102</b>
4.1	Introduction	75
4.2	Material	76
4.2.1	Collection of data and its description	76
4.3	Methodology for modeling slump of RMC using hybrid Genetic Algorithms-Artificial Neural Networks	77
4.3.1	Splitting data into training, validation, and test datasets	77
4.3.2	Data normalization	79
4.3.3	Neural network architecture and training parameters	81
4.3.4	Neural network optimization using genetic algorithms	86
4.4	Modeling slump of RMC using first order and second regressions	91
4.5	Evaluating performance of the models	93

4.6	Sensitivity analysis	96
4.6.1	Variable importance using Connection Weights Method	96
4.6.2	Response trace plots	98
4.7	Assessing applicability of the developed slump model for a different RMC batching plant	99
4.8	Developing decision support tool for estimating concrete slump	100
4.9	Summary	100
<b>Chapter 5</b>	<b>Results and Discussion</b>	<b>103-155</b>
5.1	Introduction	103
5.2	Determining the optimal number of neurons in the hidden layer	103
5.3	Genetic algorithms (GA) assisted training of artificial neural networks	107
5.3.1	Estimating optimal GA parameters	107
5.3.2	Evolving initial neural network weights and biases using GA	109
5.3.3	Training of LMBNN model using GA evolved weights and biases	111
5.4	Concrete slump models	112
5.4.1	Neural network models	112
5.4.1.1	Convergence of hybrid GA-LMBNN and conventional LMBNN models	112
5.4.1.2	Training, validation, and testing performance	115
5.4.2	Regression models	125
5.4.2.1	Training, validation, and testing performance	125

5.5	Neural network models versus regression models	126
5.5.1	Model prediction error	126
5.5.2	Model performance	128
5.5.3	Regression plots	130
5.6	Relative importance of RMC design mix constituents on the slump value	136
5.7	Response trace plots	142
5.8	Performance and reliability of the concrete slump model for different RMC batching plant	148
5.9	Decision support tool to estimate slump for the RMC design mix proportions	152
5.10	Summary	153
<b>Chapter 6</b>	<b>Summary and Conclusions</b>	<b>156-167</b>
6.1	Research summary	156
6.2	Conclusions	161
6.3	Recommendations	165
6.4	Future scope of study	166
<b>References</b>		<b>168-183</b>
<b>Appendix I</b>	<b>Design mix proportions and slump data collected from first RMC batching plant (493 datasets)</b>	<b>184-194</b>
<b>Appendix II</b>	<b>Design mix proportions and slump data collected from second RMC batching plant (100 datasets)</b>	<b>195-197</b>
<b>Publication List</b>		<b>198</b>
<b>Author's Bio-data</b>		<b>199</b>

## List of Figures

<b>Figure 1.1</b>	Mould for slump test of concrete	3
<b>Figure 2.1</b>	Biological neuron	17
<b>Figure 2.2</b>	Schematic structure of an artificial neuron and its processing mechanism	18
<b>Figure 2.3</b>	Single layer Feedforward neural network	19
<b>Figure 2.4</b>	Multilayer Feedforward neural network with single hidden layer	20
<b>Figure 2.5</b>	Multilayer recurrent neural network with single hidden layer	20
<b>Figure 2.6</b>	Neural network models and their field of application	21
<b>Figure 2.7</b>	Backpropagation neural network	24
<b>Figure 2.8</b>	Genetic algorithm cycle	31
<b>Figure 2.9</b>	Roulette wheel selection	32
<b>Figure 2.10</b>	Rank selection	33
<b>Figure 2.11</b>	Tournament selection	33
<b>Figure 2.12</b>	Stochastic sampling	34
<b>Figure 2.13</b>	Single-point crossover	35
<b>Figure 2.14</b>	Multi-point crossover	35
<b>Figure 2.15</b>	Uniform crossover	36

<b>Figure 2.16</b>	Scattered crossover	36
<b>Figure 2.17</b>	Mutation using flipping	37
<b>Figure 2.18</b>	Mutation using interchanging	37
<b>Figure 2.19</b>	Mutation using reversing	37
<b>Figure 4.1</b>	Data collection and preparing data for neural network modeling	78
<b>Figure 4.2</b>	Training of MFNN using backpropagation algorithm	85
<b>Figure 4.3</b>	Evolving weights and biases for MFNN using GA and subsequent training and validation of LMBNN	87
<b>Figure 5.1</b>	Variation of average standard deviation (AVSD) of training and validation error with hidden layer neurons	106
<b>Figure 5.2</b>	Variation of average mean absolute (AVMAE) of training and validation error with hidden layer neurons	106
<b>Figure 5.3</b>	Schematic of the optimal neural network architecture used in the study	107
<b>Figure 5.4</b>	Fitness function with varying crossover fraction ( $P_c$ ) and mutation rate ( $P_m$ )	108
<b>Figure 5.5</b>	Typical plot of fitness function versus population generations for GA	110
<b>Figure 5.6</b>	Typical plot of fitness function versus CPU time during training of GA-LMBNN model	112
<b>Figure 5.7</b>	Typical training plot for GA-LMBNN model	113
<b>Figure 5.8</b>	Typical training plot for LMBNN model	113
<b>Figure 5.9</b>	Training epochs for each independent run of the model	114
<b>Figure 5.10</b>	Training CPU time for each independent run of the model	114



<b>Figure 5.11</b>	Plot of training performance metric RMSE for each model run	118
<b>Figure 5.12</b>	Plot of validation performance metric RMSE for each model run	118
<b>Figure 5.13</b>	Plot of training performance metric MAPE for each model run	119
<b>Figure 5.14</b>	Plot of validation performance metric MAPE for each model run	119
<b>Figure 5.15</b>	Plot of training performance metric R for each model run	120
<b>Figure 5.16</b>	Plot of validation performance metric R for each model run	120
<b>Figure 5.17</b>	Plot of training performance metric E for each model run	121
<b>Figure 5.18</b>	Plot of validation performance metric E for each model run	121
<b>Figure 5.19</b>	Plot of training performance metric RSR for each model run	122
<b>Figure 5.20</b>	Plot of validation performance metric RSR for each model run	122
<b>Figure 5.21</b>	Plot of performance metric PBIAS for each run of GA model	123
<b>Figure 5.22</b>	Plot of performance metric PBIAS for each run of GA-LMBNN model	123
<b>Figure 5.23</b>	Plot of performance metric PBIAS for each run of LMBNN model	124
<b>Figure 5.24</b>	Error histogram for (a) First order regression model; (b) Second order regression model	127
<b>Figure 5.25</b>	Error histogram for (a) GA-LMBNN model; (b) LMBNN model	128
<b>Figure 5.26</b>	Percent correct data for concrete slump models	130
<b>Figure 5.27</b>	Regression plot between actual and first order regression model predicted concrete slump	134

<b>Figure 5.28</b>	Regression plot between actual and second order regression model predicted concrete slump	134
<b>Figure 5.29</b>	Regression plot between actual and GA-LMBNN model predicted concrete slump	135
<b>Figure 5.30</b>	Regression plot between actual and LMBNN model predicted concrete slump	135
<b>Figure 5.31</b>	Relative importance of the concrete design mix constituents on the slump value	139
<b>Figure 5.32</b>	Response trace plot of slump value plotted against cement content	143
<b>Figure 5.33</b>	Response trace plot of slump value plotted against PFA content	144
<b>Figure 5.34</b>	Response trace plot of slump value plotted against sand content	145
<b>Figure 5.35</b>	Response trace plot of slump value plotted against CA(20mm) content	146
<b>Figure 5.36</b>	Response trace plot of slump value plotted against CA(10mm) content	147
<b>Figure 5.37</b>	Response trace plot of slump value plotted against superplasticizer content	147
<b>Figure 5.38</b>	Response trace plot of slump value plotted against water content	148
<b>Figure 5.39</b>	Regression plot between actual and model predicted slump for second RMC batching plant	149
<b>Figure 5.40</b>	Percentage error for the data collected from second RMC batching plant	151
<b>Figure 5.41</b>	Decision support tool to estimate slump value for RMC	152

## List of Tables

<b>Table 1.1</b>	Recommended slump values as per IS:456-2000	2
<b>Table 2.1</b>	Transfer functions used in artificial neurons	23
<b>Table 4.1</b>	Details and statistics of exemplar RMC data used for modeling slump	77
<b>Table 4.2</b>	Statistical parameters of training, validation, and test datasets	80
<b>Table 4.3</b>	Details and statistics of RMC data used for validating the applicability of the developed model for concrete slump	99
<b>Table 5.1</b>	Training and validation AVSD and AVMAE for different neural network configurations	105
<b>Table 5.2</b>	CPU time and value of fitness function for different population sizes	109
<b>Table 5.3</b>	Optimum GA parameters used for the study	109
<b>Table 5.4</b>	Generations, best fitness function value and CPU time for independent runs of GA	111
<b>Table 5.5</b>	Statistics of training epochs and CPU time for independent runs of LMBNN and hybrid GA-LMBNN models	115
<b>Table 5.6</b>	Statistics of the performance metrics	116
<b>Table 5.7</b>	Performance metrics for the best GA-LMBNN and LMBNN model	124
<b>Table 5.8</b>	Performance metrics for the first order and second order regression models	125
<b>Table 5.9</b>	Statistics of the prediction error exhibited by the regression and neural network models	126
<b>Table 5.10</b>	Model performance for the overall data set	129
<b>Table 5.11</b>	Regression parameter estimates, t-value and p-value for slope	131

<b>Table 5.12</b>	Regression parameter estimates, t-value and p-value for slope	131
<b>Table 5.13</b>	Regression parameter estimates, t-value and p-value for intercept	132
<b>Table 5.14</b>	Neural network weights between input layer and hidden layer	138
<b>Table 5.15</b>	Neural network weights between hidden layer and output layer	138
<b>Table 5.16</b>	Relative importance of concrete design mix constituents on concrete slump	139
<b>Table 5.17</b>	Model performance metrics for the data set collected from second RMC plant	149
<b>Table 5.18</b>	Regression parameter estimates, t-value and p-value for slope	150
<b>Table 5.19</b>	Regression parameter estimates, t-value and p-value for intercept	151

## List of Symbols and Abbreviations

RMC	Ready Mix Concrete
BP	Backpropagation algorithm
ANN	Artificial Neural Networks
MFNN	Multilayer Feedforward Neural Network
BPNN	Backpropagation Neural Network
LMBNN	Levenberg-Marquardt Backpropagation Neural Network
GA	Genetic Algorithms
GA-LMBNN	Hybrid Genetic Algorithms-Levenberg Marquardt Backpropagation Neural Network
PFA	Pulverized fuel ash
CA(20mm)	Coarse aggregate 20mm
CA(10mm)	Coarse aggregate 10mm
SP	Superplasticizer
$H$	Hessian matrix
$J$	Jacobian matrix
$\mu$	Marquardt parameter
$p_{norm}$ , $p_{max}$ and $p_{min}$	Normalized, maximum and minimum value of variable $p$ respectively
$T_i$	Target or actual value of slump for $i$ th pattern

$P_i$	Predicted value of slump for $i$ th pattern
$\overline{P}, \overline{T}$	Mean of model predicted and actual value of slump
$N$	Number of data pairs
$e$	Network error evaluated as difference between $T_i$ and $P_i$
$\overline{e}$	Mean of $e$
$SD(e)$	Standard deviation of $e$
$AVSD$	Average standard deviation of error
$AVMAE$	Average mean absolute error
$N_{Trials}$	Number of trials
$F_{fitness}$	Fitness function
$RMSE$	Root mean square error
$y$	Dependent or response variable
$x_1, x_2, x_3, \dots, x_n$	Independent or regressor variables
$\beta_0$	Intercept
$\beta_1, \beta_2, \beta_3, \dots, \beta_k$	Partial regression coefficients
$\varepsilon$	Random error
$\beta_i, \beta_{ii}, \beta_{ji}$	Linear, quadratic and interaction coefficients
$MAPE$	Mean absolute percentage error
$R$	Correlation coefficient

$R^2$	Coefficient of determination
$E$	Nash-Sutcliff efficiency
$RSR$	Root mean square error to the observations standard deviation ratio
$PBIAS$	Percent bias
$RI_x$	Relative importance of input variable $x$
$w_{xy}$	Weight of connection from input neuron to hidden layer neuron
$w_{yz}$	Weight of the connection from hidden layer neuron to the output neuron
$W_{cement}, W_{flyash}, W_{sand}, W_{aggregate}$	Weight of concrete design mix constituent in kg
$W_{superplasticizer}, W_{water}$	Density of each concrete design mix constituent in $kg/m^3$
$\rho_{cement}, \rho_{flyash}, \rho_{sand}, \rho_{aggregate}$	
$\rho_{superplasticizer}, \rho_{water}$	Volumetric proportion of the constituent
$r_i$	
$\Delta_i$	Volumetric change in the constituent
$P_c$	Crossover fraction
$P_m$	Mutation rate

## **Chapter 1**

### **Introduction**

---

- **Background**
  - **Need for the present study**
  - **Objectives of the present study**
  - **Thesis organization**
  - **Summary**
-



## 1.1 Background

### 1.1.1 Ready Mix Concrete (RMC)

Cement concrete or commonly known as concrete, is a man-made material manufactured using key ingredients namely, cement, fine aggregate, coarse aggregate, and water. It is one of the most widely used construction material throughout the world owing not only to its wide applications but also due to its versatility, strength, affordability, and durability. Rapid growth in infrastructure has given impetus to the emergence of customized materials engineered to the requirements of the construction industry. Ready Mix Concrete (RMC) is one such customized product that is fast becoming a preferred choice among the contractors and builders. RMC refers to the concrete that is explicitly manufactured at the factory or batching plant, according to a predefined design mix for delivery to the construction site in a fresh and unhardened state by truck mounted transit mixers. RMC assures better conditions of quality control at the batching plant and provides flexibility of transporting fresh concrete to congested sites for fast and hassle free construction, coupled with material reliability and durability.

### 1.1.2 Workability of concrete

#### 1.1.2.1 Definition and importance

The concrete in a mixed state possessing plasticity is defined as Fresh Concrete. The short-term properties of fresh concrete have an enormous binding on the final properties of the concrete in the hardened state, as these govern the placing and compaction of concrete on site ultimately playing an important role in establishing its strength and durability characteristics. Workability is an important property of fresh concrete and is defined by **Mindess et al. (2003)** as “*the amount of mechanical work, or energy, required to produce full compaction of the concrete without segregation*” and therefore, governs the early age operations of placing, compacting and finishing the fresh concrete at the site. Workability depends primarily on the rheology of the cement paste and the internal friction between the aggregate particles, on the one hand, and the external friction between the concrete and the surface of formwork, on the other hand (**Li, 2011**). In a nutshell, concrete is said to be workable, if it can be formed, transported, placed, compacted and finished to its final shape and

texture with minimal mechanical effort without loss of homogeneity or with sufficient resistance to segregation. Nevertheless, in actual practice, the workability depends on the type of construction and methods used for mixing, transportation and placing. IS: 456-2000 has recommended the slump values for different placing conditions that are tabulated under:

**Table 1.1:** Recommended slump values as per IS: 456-2000

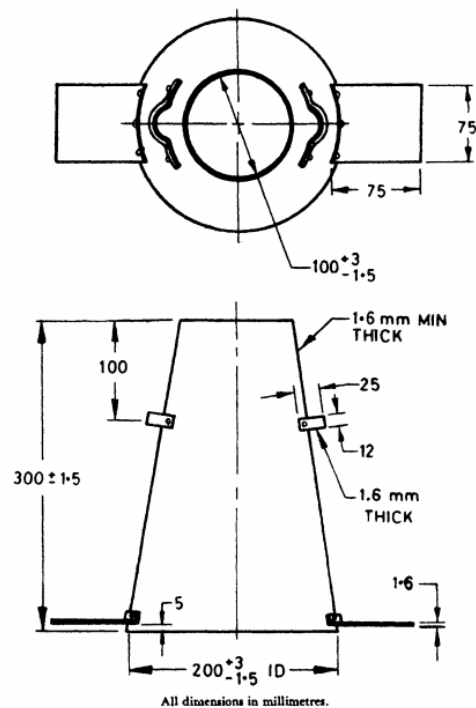
Placing conditions	Degree of workability	Slump (mm)
Blinding concrete; shallow sections; pavements using pavers	Very low	Too small to measure
Mass concrete; lightly reinforced sections in slabs, beams, walls, columns; floors; hand placed pavements; canal lining; strip footings	Low	25-75
Heavily reinforced sections in slabs; beams, walls, columns; slip formwork; pumped concrete	Medium	50-100 75-100
Trench fill; in-situ piling	High	100-150
Tremie concrete	Very high	Too large to measure

Apart from being an important property of fresh concrete necessary to achieve full compaction leading to attainment of specified strength and durability, the workability of concrete plays an equally important role in the success of the RMC industry. It not only ensures consistency of quality and uniformity of concrete from batch to batch but also, helps in ascertaining the maximum transit time that a fresh ready mix concrete can undertake from manufacturing at the batching plant, subsequent transportation and placing on the construction site without allowing

stiffening of concrete paste. Moreover, since workability of concrete is governed by the type of construction namely, heavily reinforced sections, lightly reinforced sections, road pavements, shallow sections or construction requiring intensive vibration, demanding high, medium, low, very low or extremely low workability concrete respectively, it helps in formulating the design mix for concrete catering to the construction requirements. The workable concrete also eases the pumping of the concrete at the construction sites and, therefore, requires a cohesive concrete mix having sufficient resistance to segregation during transportation and placing.

### 1.1.2.2 Workability assessment: Slump test

Workability of concrete is a combination of two basic properties of fresh concrete, namely, consistency and cohesiveness. Consistency defined as the ease with which fresh concrete can flow whereas cohesiveness refers to the water and coarse aggregate holding capacity of fresh concrete. Since workability is defined as the mechanical work required for placing and compaction of fresh concrete and is dependent on the type and method of construction, there no single well-accepted test for measuring workability. However, in quantitative terms, the consistency of concrete is measured using a widely used test called the Slump Test. The test consists of mould resembling frustum of a cone (**Figure 1.1**).



**Figure 1.1:** Mould for slump test of concrete (Source: IS: 7320-1974)

The mould is filled with three layers of fresh concrete. The concrete is compacted, and the mould is lifted, allowing the unsupported cone of fresh concrete to collapse under its weight. The decrease in the height of the cone correct to 5 mm is measured and designated as the slump value of fresh concrete. Although slump reflects the yielding of concrete under its weight yet, it is a useful site test that can be used to monitor batch-to-batch concrete mix and the resulting variation in workability. A true slump exhibited by slumping of concrete all around indicating a cohesive mix, whereas if the concrete slumps along an inclined plane, a shear slump is noticed indicating a harsh mix. In a mix having stiff consistency, a zero slump is noticed.

### 1.1.2.3 Factors affecting workability

The workability of concrete is affected by every component of the concrete and essentially every condition under which it was produced. It is seen that water content, cement content, aggregate characteristics, and admixture dosage influence the consistency and cohesiveness of concrete, thereby affecting the workability of concrete. The manner in which these factors affect the workability of concrete is discussed in the following sub-sections.

- (i) **Water Content:** The water content is the most important factor influencing the workability of concrete as it fills the voids between the particles and acts as a lubricating agent decreasing the intermolecular friction. The water content thus results in the fluidity of the fresh concrete making it easy to be compacted. However, at higher water content, the concrete mix tends to lose its cohesiveness leading to adverse effects of segregation and bleeding. Higher water content also reduces the strength of concrete.
- (ii) **Cement content:** The cement content has a dual effect on the workability of concrete. For a given water-cement ratio, it tends to increase the water content in the concrete at larger cement content thereby, aiding the consistency of concrete. The fine particles of cement coat the aggregate particles and fill the void spaces, assisting the lubrication of aggregate particles. Low cement content produces harsh mixes producing concrete of poor consistency. However, an increase of cement content at same w/c ratio

tends to increase the workability, but a very high proportion of cement may lead to mixes that tend to be sticky.

- (iii) **Aggregate characteristics:** Fine sand in the concrete increases the water demand for a given consistency leading to harsh mixes that are unworkable. The size of coarse aggregates influences the workability of concrete. For the same volume, the larger sized coarse aggregates have a smaller surface area than smaller aggregates, requiring lesser paste for coating. Hence, in the case of larger coarse aggregates, more amount of paste is available for lubricating the particles, thus enhancing the workability.
- (iv) **Admixture Dosage:** Both chemical and pozzolanic admixtures influence the workability of concrete. Pozzolanic admixtures tend to improve the cohesiveness of concrete due to the spherical shape of the particles and glassy surface. Superplasticizers are chemical admixtures added to the concrete to increase its workability at a given water-cement ratio and to keep the fresh concrete in a green state for a longer duration of time.

### 1.1.3 Soft Computing Techniques

Coined by Lotfi A. Zadeh in the year 1994, Soft Computing techniques differ from the hard computing counterparts, as they tackle and provide low-cost solutions to practical problems based on basic intelligence, reasoning, common sense and ability to create analogies and approaches, synonymous with the decision-making ability inherent in humans. The ideology behind Soft Computing approaches is to create a synergy between distinct artificial intelligence techniques and to bring them under one umbrella.

The synergistic approach facilitates these techniques to deal with real life problems that inherently possess partial, vague, noisy and incomplete information. The Soft Computing employs a variety of statistical, probabilistic and optimization tools, to provide low cost and tractable solutions to the problems encountered in real life where direct expertise is absent. Among the various components of Soft Computing, Artificial Neural Networks, Genetic Algorithms, and Fuzzy Logic have found greater importance and applicability. In the present study, two Soft Computing

techniques namely Artificial Neural Networks and Genetic Algorithm are harnessed for developing a mathematical model for the concrete slump.

### 1.1.3.1 Artificial Neural Networks: A modeling tool

The different nature, type, and properties of the constituents used in the concrete design mix impart a composite nature to concrete, making mathematical modeling of its properties a highly complex and non-linear function of the design mix constituents. Many widely accepted empirical relationships derived from experimental data are available in the form of regression equations and are commonly in use for extracting knowledge about properties of concrete. These do not provide the expected accuracy and predictability when there are a number of independent and dependent variables whose interactions are either unknown, non-linear or difficult to represent. Artificial Neural Networks (ANN) touted as the next generation of computing has provided a suitable substitute to the conventional mathematical modeling of the real life phenomenon influenced by vague, imprecise and uncertain information.

Inspired by the working of a human brain, ANN learns through parallel processing of information from the historical data without requiring prior specific knowledge about the interactions among and between the independent and dependent variables. Some notable recent applications of ANN for material modeling of concrete include, compressive strength prediction models for High Performance Concrete containing nano silica and copper slag (**Chithra et al., 2016**), plasticity model for concrete in compression (**Briki and Djeghaba, 2015**), prediction of compressive strength of concrete due to sulphate attack (**Diab et al., 2014**), strength prediction of High Strength concrete (**Tayfur et al., 2014**), modeling chloride diffusivity in high performance concrete (**Hodhod and Ahmed, 2013**), prediction of compressive strength of recycled aggregate concrete (**Duan et al., 2013**), predicting drying shrinkage of concrete (**Bal and Buyle-Bodin, 2013**), predicting properties of high performance concrete containing cementitious materials (**Khan, 2012**), prediction of core compressive strength of self-compacting concrete (**Uysal and Tanyildizi, 2011**), predicting compressive strength of self compacting concrete containing bottom ash (**Siddique et al., 2011**), and predicting strength of rubberized concrete (**Abdollahzadeh et al., 2011**).

### 1.1.3.2 Genetic Algorithms: An optimization tool

The problems faced in real life are multi-faceted, having numerous feasible solutions. The difficulty in choosing the best solution has diverted the attention of the researchers to learn from nature. The “*survival of the fittest*” paradigm propounded by Charles Darwin, finds its usefulness in devising algorithms for finding the fittest individual that can survive the environmental extremes and produce offspring. Based on the ideas of the biological evolution and genetics, a population-based stochastic search algorithm known as Genetic Algorithm (GA) developed by John Holland, is now one of the most widely used global search and optimization algorithms. The ability of GA to simultaneously search through a number of solutions and gradually improve the quality of the solution to reach an optimum solution has opened up new avenues for development of time-saving techniques that can replace the conventional design and optimization procedures. Some of the recent applications of GA related to concrete include, optimization of concrete shells (**Bertagnoli *et al.*, 2014**), cost optimization of pre-stressed concrete bridges (**Aydin and Ayvaz, 2013**), mix proportioning of recycled aggregate concrete (**Park, 2013**), optimized design of reinforced cantilever wall (**Pei and Xia, 2012**), and cost optimization based design of precast concrete floors (**Albuquerque *et al.*, 2012**).

## 1.2 Need for the present study

Ready Mix Concrete (RMC) is an engineered product manufactured at the batching plant that, aims to customize compressive strength and workability of concrete mix for the construction requirements. Regardless of the technological advancements in RMC industry, the concrete mix that cannot be easily placed or compacted at the site is not likely to yield the expected strength and durability characteristics. The evaluation of concrete workability in terms of initial slump value at the batching plant helps in ascertaining the maximum transit time that an RMC can undertake up to the construction site, without considerable loss in workability and for ensuring that the RMC delivered at a site has a customized workability satisfying a particular type of construction. Therefore, in spite of providing premium grade concrete catering to the structural requirements, the importance of the workability in RMC industry cannot be undermined.

The procedure for trial mix design is a tedious process, as it involves multiple trials with different proportions of concrete constituents. Although, experimental procedures in any industry cannot be ruled out as they form the backbone of further research and development, yet a decision support tool can surely help to quickly estimate concrete slump for a particular design mix and further provide a fair idea to the technical personnel regarding design mix proportions for the desired workability. A decision support tool for concrete design mix, harnessing the knowledge extracted from the past results of the concrete mix designs is, therefore, essential for easing the burden of performing regular multiple trials with different design mix proportions. Apart from developing a decision support tool for the concrete slump, a mathematical model is also necessary to provide insight into the complex material behavior of concrete.

The different nature, type, and properties of the constituents used in the concrete design mix impart composite nature to concrete, making Multilayer Feedforward neural networks (MFNN) trained using backpropagation (BP) algorithm also known as backpropagation neural networks (BPNN) a preferred choice for modeling its material behavior as shown in (Alshihri *et al.*, 2009; Atici, 2011; Deshpande *et al.*, 2014; Diab *et al.*, 2014; Ghafoori *et al.*, 2013; Kostic and Vasovic, 2015; Mermerdas and Arbili, 2015; Sbartai *et al.*, 2009; Sobhani *et al.*, 2010; Zavrtnik *et al.*, 2016). Moreover, in all previous literature (Chine *et al.*, 2010; Dias and Pooliyadda, 2001; Jain *et al.*, 2008; Oztas *et al.*, 2006; Yeh, 2006; Yeh, 2007; Yeh, 2008), the researchers have harnessed the BPNN for modeling the slump of concrete. The reason for this wide use of BPNN is its ability to imbibe the non-linear or unknown input-output relationships easily through systematic updating of neural network weights and biases.

In spite of its simple implementation, the BP algorithm is faced with some inherent drawbacks. BP algorithm is a local search algorithm that harnesses the principle of steepest gradient descent for minimizing the error function, evaluated preferably as the squared error between the actual and the predicted outputs. Since, the BP algorithm updates the weights and biases by evaluating the gradient of the error function, it is not suitable for approximating functions that are discontinuous and non-differentiable (Asadi *et al.*, 2013; Rocha *et al.*, 2014). The BP algorithm initialized with a random draw of weights and biases, forces the neural network to behave



differently during each re-run of the neural network training, providing inconsistent performances (**Sexton et al., 1999**). The local search BP algorithm follows the trajectory of steepest descent to find the optimal solution (**Sutton, 1986; Whitley et al., 1990**) and is, therefore, likely to get trapped in local minima and exhibit slow rate of convergence (**Sahoo and Maity, 2007**).

Amongst the numerous improved backpropagation algorithms introduced in the past namely, Resilient Backpropagation, BFGS Quasi-Newton, Scaled Conjugate Gradient, etc., the Levenberg-Marquardt (LM) backpropagation algorithm is the most efficient and fastest converging algorithm (**Hagan and Menhaj, 1994**) and has been preferred over other backpropagation optimization algorithms for updating the neural network weights and biases (**Daliakopoulos et al., 2005; Dee et al., 2011; Kisi, 2008; Paliwal and Kumar, 2011**). Although, LM algorithm inherits faster convergence of the Gauss-Newton algorithm yet, it still carries the weakness of the gradient descent algorithm, heavily depending on the initial draw of weights and biases to avoid entrapment at local minima and subsequently converge to global optimum (**Kermani et al., 2005; Nawi et al., 2013**). The probability of initial weights and biases located on a local minimum of the error function increases the likelihood to converge to local minima (**Sexton and Dorsey, 2000**). The values of weights and biases used for initializing the backpropagation algorithm are therefore required to be optimized to guarantee its convergence to the global optimum.

In comparison to local search BP algorithm that requires gradient information for updating of weights and biases, the GA employs gradient free global search. Although this search is global in nature yet, it is susceptible to slow convergence owing to weak local search ability (**Ghaffari et al., 2006; Kitano, 1990**). GA being a global search heuristic promotes both exploitation and exploration of the search space as against localized exploitation of the search space rendered by BP algorithm. Genetic Algorithm's non-trajectory search and exploratory power give an edge to the GA algorithm over the BP algorithm to perform longer jumps in search space thereby, reducing the probability of entrapment at local minima (**Mavrovouniotis and Yang, 2015**).

It is seen that the individual soft computing techniques namely GA and BP algorithm have certain inherent advantages and drawbacks concerning training an MFNN. Hence, there is a need for developing a hybrid methodology for training

MFNN, with each technique complementing each other, helping to cover up the individual drawbacks thereby, leading to the development of a robust computational tool. In the light of the advantages rendered by hybridization and the limitations of the original procedures, a computational model for the concrete slump based on proportions of RMC design mix constituents has been attempted by hybridizing the global exploration ability of GA, utilized for evolving the optimal set of initial weights and biases with, the fast convergence rendered by the LM backpropagation algorithm, for assisting the MFNN trained using LM backpropagation algorithm (LMBNN) to minimize entrapment at local minima and accomplish faster convergence.

### **1.3 Objectives of the present study**

For the present study “Modeling and Analysis of Ready Mix Concrete Slump using Hybrid Genetic Algorithms-Artificial Neural Networks”, following research objectives are identified:

- (i) Model the functional relationship between proportions of Ready Mix Concrete design mix constituents and slump value.
- (ii) Compare the effectiveness of Genetic Algorithms, Levenberg-Marquardt Backpropagation Algorithm and the hybrid Genetic Algorithm-Levenberg Marquardt Backpropagation Algorithm for training MFNN, in terms of prediction consistency and accuracy and speed of convergence.
- (iii) Compare the potential of neural network methodology and conventional regression techniques for modeling the material behavior of concrete slump.
- (iv) Assess the relative importance of design mix constituents of Ready Mix Concrete on the slump value.
- (v) Utilize the developed slump model to provide insight into the material behavior exhibited by the concrete slump.
- (vi) Explore the effectiveness and applicability of the developed model for predicting slump value for the design mix proportions of a different RMC batching plant.

- (vii) Harness the slump model to develop a decision support tool that aids prediction of slump for the given concrete design mix proportions.

#### **1.4 Thesis organization**

The thesis has been divided into six chapters. A brief overview of the contents of each chapter is summarized as under:

- Chapter 1:** The chapter introduces the theoretical background of all the components involved in the research work. The need for the present study and the research objectives has been dealt in this chapter.
- Chapter 2:** This chapter introduces the soft computing methodologies used in the study namely, Artificial Neural Networks, and Genetic Algorithms and further provides a detailed description regarding the components and the working of each methodology.
- Chapter 3:** An extensive literature survey of the past research has been covered in this chapter. The literature survey discusses the applications of artificial neural networks for modeling the various properties of concrete including the slump value. The chapter also highlights the various multidisciplinary applications and discusses the civil engineering applications of the hybrid genetic algorithm-artificial neural networks methodology. Based on the inference drawn from the literature survey, the research gap is identified.
- Chapter 4:** The chapter elaborately discusses the methodology used for modeling the slump of Ready Mix Concrete. The chapter incorporates collection, division, and normalization of data, determination of neural network architecture, optimization of neural networks using genetic algorithms, evaluation of the performance of models, comparison of the neural network model with conventional regression models, sensitivity analysis, applicability of the slump model for a different RMC batching plant and development of decision support tool for prediction of slump value.

- Chapter 5:** The chapter highlights the results of the study and further discusses them at length. The results related to finding optimal the neural network architecture, computing the optimal GA parameters, training, validation and performance evaluation of hybrid GA-LMBNN and its comparison with conventional LMBNN model, comparison of neural networks with regression techniques, sensitivity analysis, performance of the developed slump model for predicting concrete slump for the design mix data collected from a different RMC batching plant and decision support tool for concrete slump have been highlighted and discussed.
- Chapter 6:** The chapter summarizes the study and presents the conclusions drawn from the study and recommendations for future work.
- Appendix I:** Ready Mix Concrete data showing the proportions of different design mix constituents and corresponding slump values collected from the first RMC batching plant.
- Appendix II:** Ready Mix Concrete data showing the proportions of design mix constituents and corresponding slump values collected from the second RMC batching plant.

## **1.5 Summary**

Ready Mix Concrete (RMC) has emerged as a preferred construction material owing to its premium quality, customized to the type of construction. The workability of concrete measured using slump test at RMC batching plant is an important quality assurance parameter and helps in ascertaining that the RMC delivered at the site is of the desired workability.

The procedure of designing a concrete mix catering to customized workability is tedious as it requires repeated trials with different proportions of constituents. Quick estimation of concrete slump without undergoing cumbersome experimental procedures necessitates the development of a decision support tool. Apart from this, a mathematical model that can provide insight into the complex material behavior of concrete slump is also necessary.

Concrete is a composite material and, therefore, its material modeling demand a non-algorithmic approach. Artificial Neural Networks (ANN), a subset of Soft Computing, has been in vogue for the last few decades for tasks associated with modeling material behavior. ANNs mimics the working of the human brain and learns through parallel processing of information from the historical data without requiring prior specific knowledge about the interactions among and between the independent and dependent variables. Amongst the numerous available architectures for ANN, the Multilayer Feedforward Neural Network (MFNN) trained using Backpropagation (BP) algorithm is widely preferred by the researchers for tasks associated with material modeling of concrete.

Although BP algorithm harnessing the principle of steepest descent is simple to implement yet, it carries the drawback of slow convergence and likelihood of entrapment at local minima. Amongst the numerous improved local search backpropagation algorithm, the Levenberg-Marquardt (LM) algorithm is the fastest and the most efficient learning algorithms for training the MFNN. The LM algorithm derives its convergence speed from the Gauss-Newton method but carries the drawback of steepest descent algorithm, relying on the initial draw of neural network weights and biases to escape the local minima. Genetic Algorithms (GA) on the other hand employ global search and exploratory power to perform longer jumps in search space to evade entrapment at local minima. Although this search is global in nature yet, it is prone to slow convergence owing to large search space.

A computational model for the concrete slump based on proportions of RMC design mix constituents has been proposed by hybridizing the global search of GA for evolving the optimal set of initial weights and biases with the fast convergence rendered by the LM backpropagation algorithm, for assisting the MFNN trained using LM backpropagation algorithm (LMBNN) to escape entrapment at local minima and accomplish faster convergence. In the light of the adopted methodology and necessity of the present study, the objectives of the research have been finalized.

## **Chapter 2**

### **Artificial Neural Networks and Genetic Algorithms: An Overview**

---

- **Introduction**
  - **Artificial Neural Networks**
  - **Genetic Algorithms**
  - **Summary**
-

## **2.1 Introduction**

With the advent of computers and rapid advancements in hardware and software, there has been a persistent endeavor to integrate the human consciousness, wisdom, intuition and the ability into the computing environment for dealing with problems arising from real life's uncertainties. This nature inspired intelligence is conceived algorithmically in a computing environment by employing statistical, probabilistic and optimization tools placed under the umbrella of a multi-disciplinary field called the Soft Computing. It is an emerging field that has synergistically amalgamated the characteristics of biological systems to provide the impetus for the development of intelligent and wiser machines, catering solutions to a wide range of real-world problems influenced by uncertainty and imprecision. The systems born out of Soft Computing render human-like expertise in a particular problem domain, capable of learning and adapting themselves to the change in the problem scenario.

In contrast to hard computing relying on predefined rules, the Soft Computing is tolerant of imprecision, uncertainty, partial truth, and approximation (Zadeh, 1994) and render cost-effective and robust solutions to problems that are difficult to be solved using conventional computing techniques. The field of the Soft Computing brings forth the elements of Fuzzy Logic, Neural Computing, Evolutionary Computation, Machine Learning and Probabilistic Reasoning under one common umbrella which, either working on their own or complementing each other are able to provide practical solutions, tolerant to uncertainty and imprecision through, emulation of learning skills and cognitive ability of the human mind. Artificial Neural Networks (ANN), Genetic Algorithms (GA) and Fuzzy Logic (FL) represent the major subsets of Soft Computing. The chapter comprehensively discusses the two Soft Computing methodologies used in the study, namely Artificial Neural Networks and Genetic Algorithms.

## **2.2 Artificial Neural Networks**

Artificial Neural Networks derive their inspiration from interdisciplinary subjects, namely neuroscience, mathematics and computer science. The ideas from these fields were extracted and amalgamate to develop computational systems that

were able to learn from examples to perform the tasks intelligently. A neural network thus exploits the ability of the human brain to acquire and store information and adapt to the changing scenario just as human beings habituate themselves to the environmental extremes. The functioning of neural networks is a simplified abstract of the human brain, as they comprise of massively parallel highly interconnected artificial neurons that perform non-linear computations and connected by adaptive or tunable parameters that continuously change based on the information presented during the learning phase. These advantages have created immense interest of the researchers in the field of neural networks for dealing with problems that are ill-conditioned, containing noisy or incomplete data and related to real life dynamic environments.

### **2.2.1 Historical background**

The complexity of human brain attributed to a network of roughly hundred billion neurons with each neuron having about 10,000 synaptic linkages and its vaguely understood working has fascinated the human civilization since ages. As far as the history goes, the Greek philosophers Plato (427-347 B.C.) and Aristotle (384-322 B.C.) were the first to correlate the structure of the human brain with its functioning. As the time passed by, several developments were made in the field of neuroscience. Continuous research performed by Benjamin Franklin, Charles Bell, Gustav Theodore Fritsch and Karl Wernicke, created an insight into the complex structure and the nature of the human brain that strengthened the foundations of neuroscience. A breakthrough in the field of neuroscience was achieved in the early part of the twentieth century through research conducted by Camillo Golgi and Santiago Ramon y Cajal, which culminated in their sharing the sixth Nobel Prize for medicine in 1906.

The idea of incorporating the learning abilities of the human brain in the computing environment started in 1943 when, neurophysiologist Warren McCulloch and Walter Pitts conceptualized a *Boolean brain*, which can be considered as the first neural network. In 1949, Donald O. Hebb published a book *The Organization of Behavior* which introduced the word *connectionism* related to the model of the human brain. It provided the first insight into the biological neuron's learning mechanism through a change in the strength of the synaptic connections. The research progressed



further with Ross Ashby book named *The Design for a Brain* published in 1954. In 1958, Frank Rosenblatt introduced the idea of Perceptron, which was a refined and a more sophisticated version of an artificial neuron developed by Warren McCulloch and Walter Pitts. A significant research was put in place with Perceptron learning for tasks associated with pattern recognition.

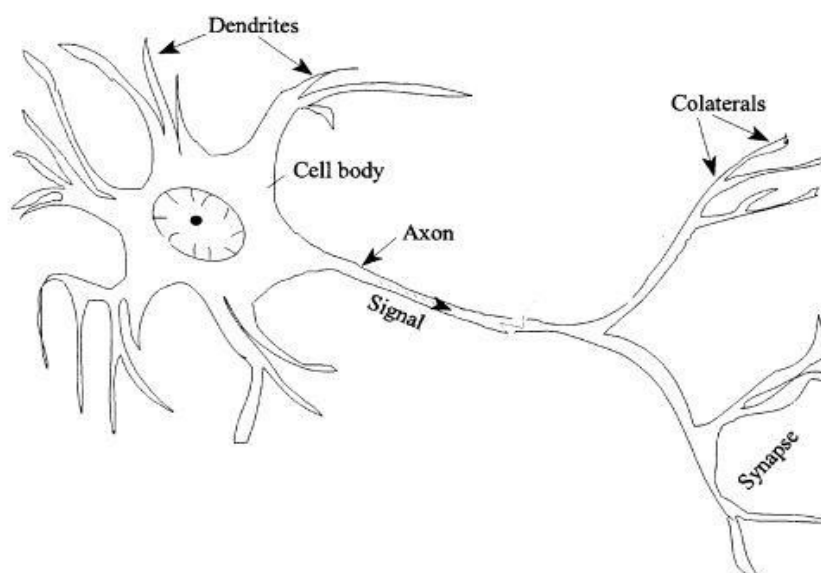
The following decade 1959-1969 showed a decreased interest in the field of Neural Networks. The interest in the field of Artificial Intelligence was revived by Minsky and Papert through their book *Perceptrons: An Essay in Computational Geometry* but still the research related to neural networks went into the background. Teuvo Kohonen with the development of Self-Organizing Maps (SOM) in 1982 and subsequently in the same year John Hopfield and Tank developed the Hopfield network, to bring the neural network research back on track. However, major credit for the revival of interest in neural networks goes to Rumelhart, Hinton and Williams for discovering the backpropagation algorithm in 1986. The backpropagation algorithm is still the most commonly used learning algorithm in neural networks.

With subsequent developments in the field of computer hardware and software, different types of neural network models came into limelight. Adaptive Resonance Theory (ART) was put forward by Carpenter and Grossberg in 1987. The Radial Basis Function (RBF) neural network for tackling problems related to multivariate interpolation was created by Broomhead and Lowe in 1988. The work on character recognition was presented by Fukushima in 1988 with the advent of Neo cognition neural networks.

### **2.2.2 Basic functional unit: An artificial neuron**

ANN emulates the functioning of the human brain and contains basic functional groups called the neurons, nodes or artificial neurons. The artificial neuron-inspired by the structure and operation of a biological neuron represents a simplified abstract of its biological counterpart and therefore before going into the mathematics and computational part of an artificial neuron; it is pertinent to grasp a brief overview of the biological neuron. A biological neuron forms the core processing unit in a brain and is made up of roughly spherical shaped cell body called the soma. The soma contains many irregularly shaped branched filaments called dendrites. The inputs or signals from other neighboring connected neurons enter the soma through synaptic

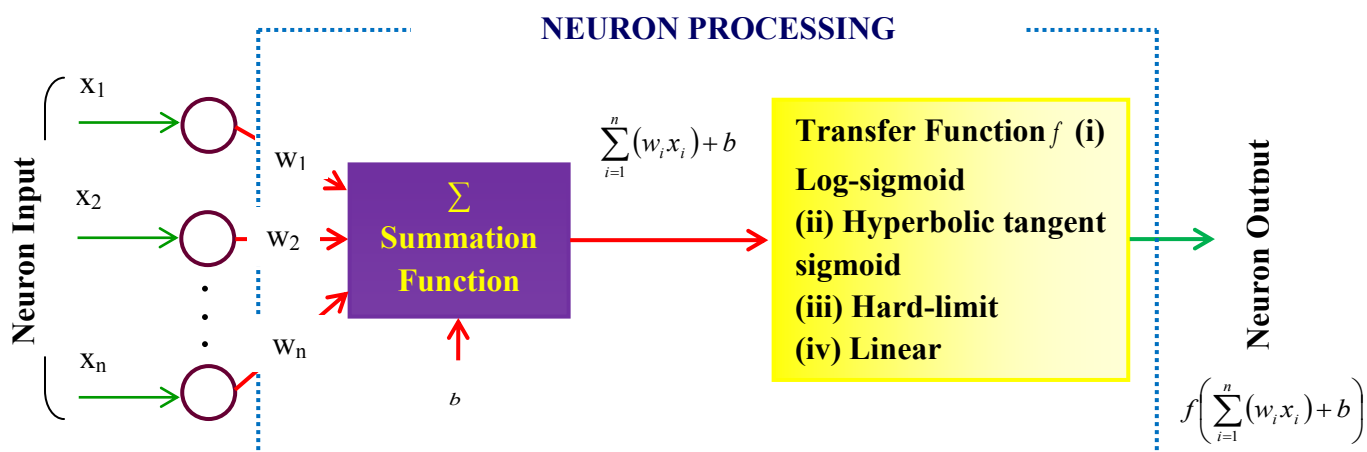
connections located on the dendrites. The soma performs the function of combining these signals or inputs. If the resultant signal is found to be greater than a predefined threshold value, it fires an electrical impulse which is further communicated to other neurons through a long extension of the cell body called the axon. The axon transmits this output to the other neurons through multiple branches, each terminating in a synapse. The synapse provides a weighted electrical connection between the two neurons, which based on the strength of the impulse, received either, passes excitatory or inhibitory impulses to the next connected neuron. The structure of a biological neuron is shown in **Figure 2.1**.



**Figure 2.1:** Biological neuron (Source: Basheer and Hajmeer, 2000)

An artificial neuron is a mathematical model which attempts to capture the essential features of the biological neurons namely, controlling the strength of incoming signals through dendrites  $x_1, x_2, \dots, x_n$  by applying a multiplication factor called the synaptic weights  $w_1, w_2, \dots, w_n$  and applying the summation function to combine the weighted signals. Additional weight, called the bias  $b$  is also added to the weighted output of a neuron to render its rapid convergence through the shifting of origin of a transfer function. Moreover, the incorporation of a bias allows a neuron to provide an output even if its input is zero. The resultant weighted sum of

inputs and bias  $\sum_{i=1}^n (w_i x_i) + b$  is finally acted upon by a threshold function called the transfer function or activation function to produce the output of a neuron  $f\left(\sum_{i=1}^n (w_i x_i) + b\right)$ . **Figure 2.2** shows the schematic structure and processing of an artificial neuron.



**Figure 2.2:** Schematic structure of an artificial neuron and its processing mechanism

### 2.2.3 Characteristics

A neural network is characterized by three fundamental entities namely, the arrangement of neurons and their connections, called its architecture or topology, learning algorithm for updating or adjusting the weights associated with connections and the transfer functions related to the neuron layers.

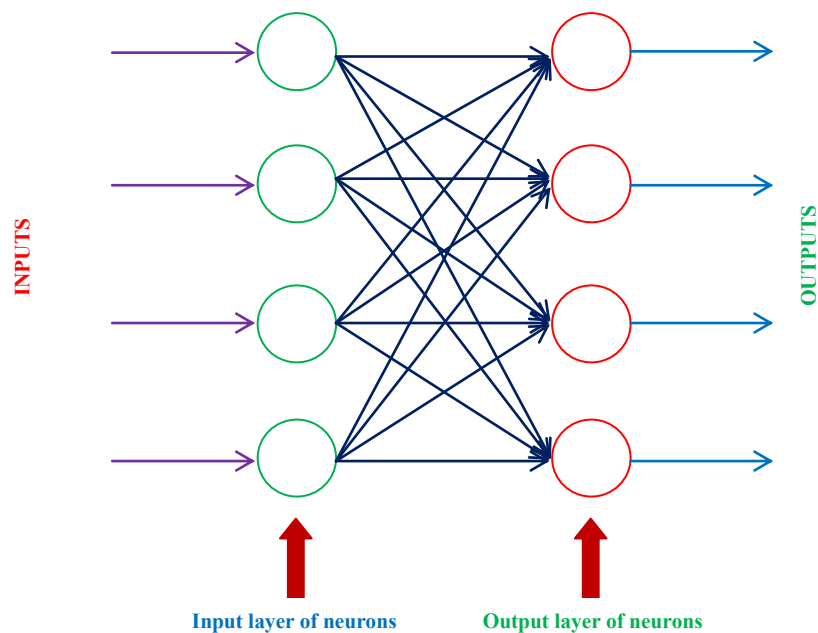
#### (i) Architecture

The processing units of ANN namely, the artificial neurons are arranged in layers and form a massively interconnected structure. The ability to process information is therefore largely dependent on the arrangement of the neurons and their geometry of interconnections. Based on these important considerations, the neural network architecture can be broadly classified into three fundamental classes:

- a) Single layer Feedforward neural networks.
- b) Multilayer Feedforward neural networks.
- c) Recurrent neural networks

As the name suggests, a Feedforward Neural Network contains neurons arranged in layers that are connected in the forward direction only, i.e. no intra-layer connections or feedback loops are permitted. This arrangement compels the information to flow in the forward direction and, therefore, their output is dependent entirely on the input provided. These networks can comprise of the input neuron layer directly connected to the layer of output neurons through a number of weighted connections leading to the formation of Single layer Feedforward Neural Networks (**Figure 2.3**) or the network can have one or more intermediate hidden layer of neurons sandwiched between input and output layer of neurons to form Multilayer Feedforward Neural Networks (**Figure 2.4**).

In the case of Feedback or recurrent neural networks (**Figure 2.5**), the signal or the information can travel in both forward as well as backward directions through neuron layers. The outputs are generated by presenting the inputs in the usual manner and further these outputs are fed back as inputs which make the neural network behave in a dynamical way, continuously change their state till attaining a state of equilibrium. In contrast to Feedforward neural networks, the recurrent neural networks exhibit short-term memory as their output state at any given time depends on the previous states. A recurrent neural network does not have any particular structure (**Yeganaryana, 2001**).



**Figure 2.3:** Single layer Feedforward Neural Network

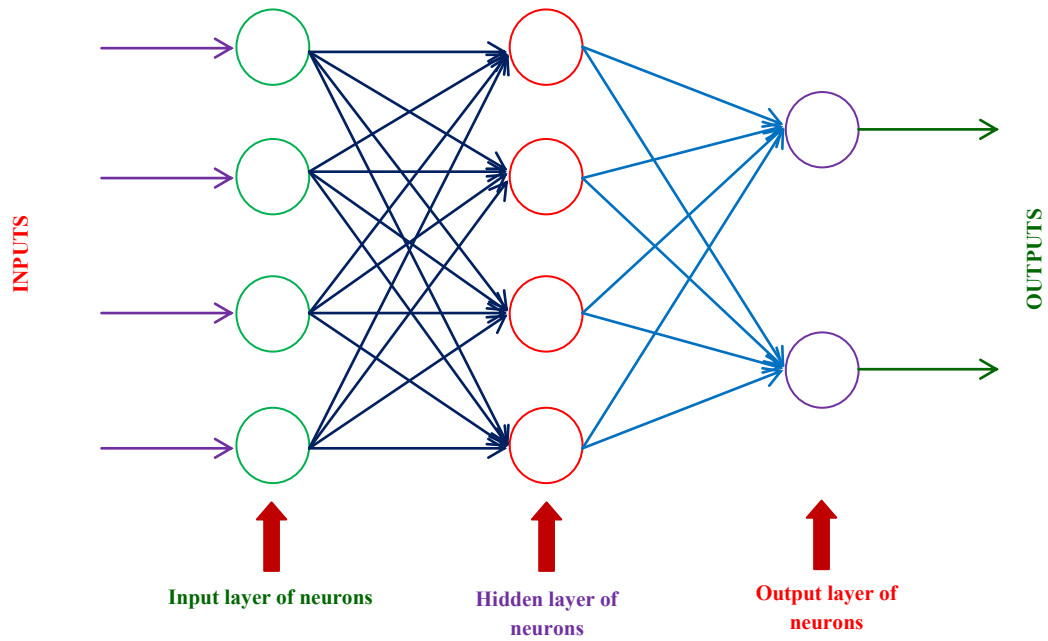


Figure 2.4: Multilayer Feedforward Neural Network with single hidden layer

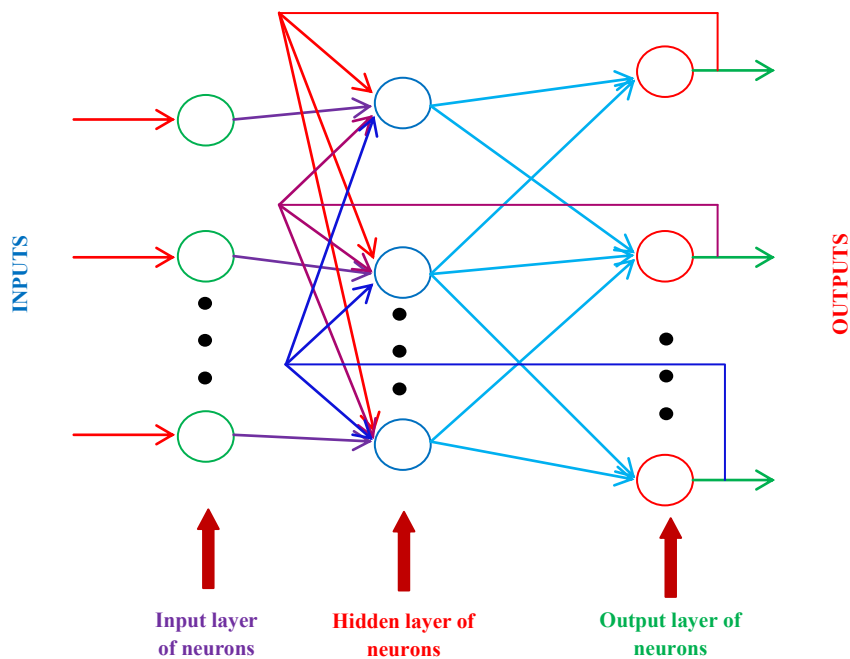
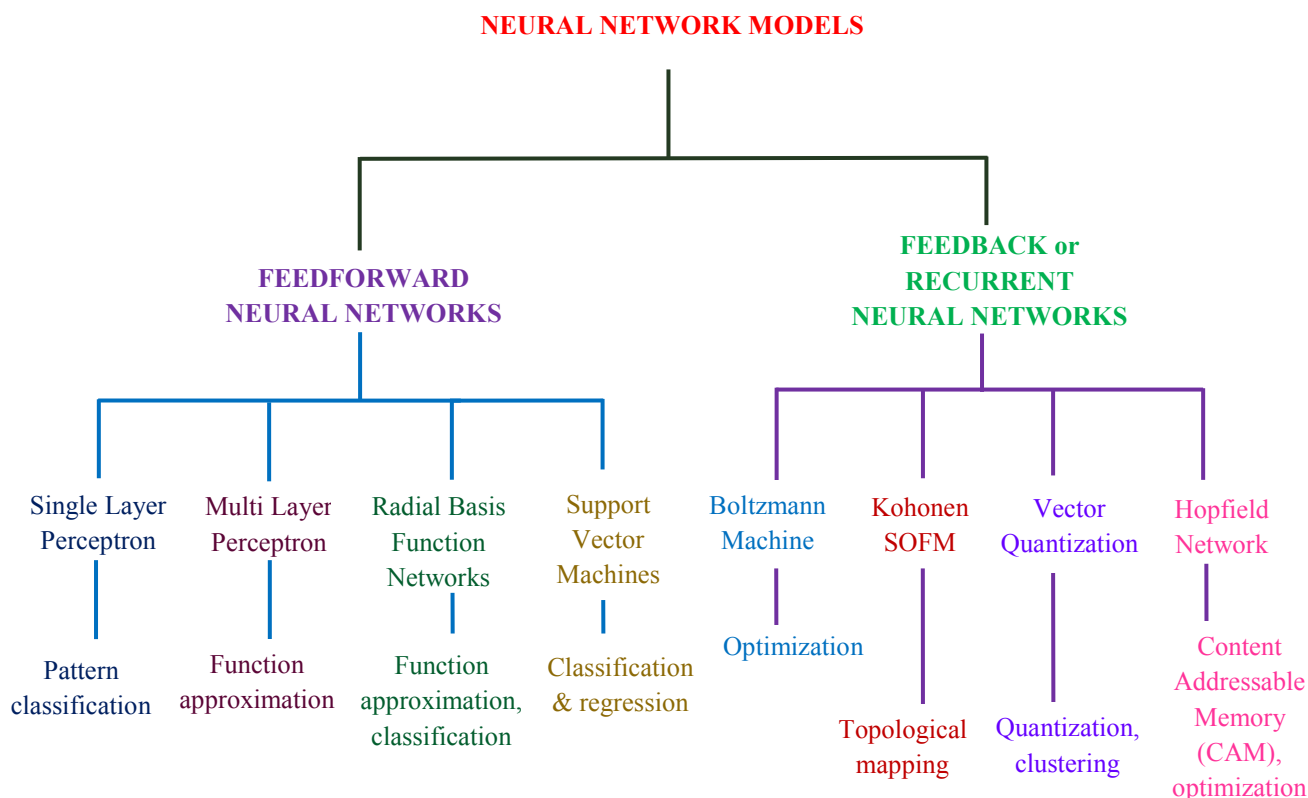


Figure 2.5: Multilayer recurrent neural network with single hidden layer

Various neural network models or architecture have been developed by different researchers to cater to the requirements of the problem in hand. **Figure 2.6** gives a hierarchical overview of various neural network models and their field of application.



**Figure 2.6:** Neural network models and their field of application

**(ii) Learning paradigms**

The successful learning or imbibing of a particular problem by ANN attributes to the information stored in the inter-neuron synaptic connections. The learning of neural networks is accomplished by a systematic updating of weights and biases to enable neural networks to predict output near the actual values. The learning paradigms in ANN can be classified as supervised, unsupervised and reinforcement. In the case of supervised learning, the information is presented to the neural network in the form of input-output data pairs with each input associated with the output. The learning rule is then applied for adjustment of weights and biases to render network error between actual or target values and neural network predicted outputs. Multilayer perceptrons, support vector machines, and radial basis function classifiers use supervised learning. In contrast to the supervised learning which utilizes a teacher or supervisor to classify the data into classes and further utilizes the class information to update the weights and biases, an unsupervised learning performs these activities heuristically without seeking supervisory assistance. Unsupervised learning is

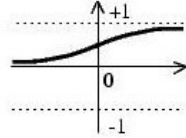
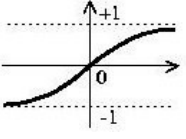
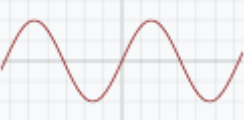
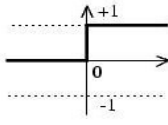
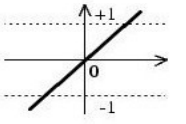
commonly used for problems associated with clustering of data into a number of similar groups. Self-organizing maps harness the unsupervised learning paradigm. Reinforcement learning uses trial and error process through adapting and exploring the features associated with the structured input pattern without requiring correct target values for each input.

The learning in ANN is based on a rule that determines the iterative procedure of updating or adjusting of free parameters namely, weights and bias over a number of training or learning cycles. **Haykin (2009)** has suggested four different types of learning rules namely, correlation learning or Hebbian learning, error-correction learning (ECL), Boltzmann learning and competitive learning. For the present study, we will restrict our discussions to error correction learning (ECL). The ECL rule used for the supervised learning of ANN modifies the weights and biases during each training cycle to reduce the arithmetic difference between the actual or target values and network predicted values to a threshold minimum. One of the most popular ECL rule-based algorithms is the error backpropagation algorithm or simply a backpropagation algorithm. The Multilayer Feedforward Neural Networks (MFNN) or Multilayer perceptron (MLP) trained using backpropagation (BP) algorithm is the most widely used neural networks and are considered the workhorse of ANNs (**Rumelhart *et al.*, 1986**). The MFNN trained using the BP algorithm commonly known as Backpropagation Neural Networks (BPNN) comprise of three basic layers of neurons namely, an input layer, an output layer and a number of intermediate hidden layers. For a particular problem, the input layer of neurons represents the input or independent variables and the output layer neurons are synonymous to the output variables or dependent variables being modeled on the basis of the dependent variables. The inclusion of the number of hidden layers and hidden layer neurons increases the complexity of the neural network, making it capable of dealing with complex and non-linear nature of problems.

### **(iii) Transfer functions**

The transfer function or the activation function typically used in neural networks are non-linear functions (log-sigmoid and hyperbolic tangent sigmoid), sinusoid, step functions (hard limit) and linear functions (**Table 2.1**) and are introduced to imitate the nonlinear characteristics of the biological neurons.

**Table 2.1:** Transfer functions used in artificial neurons

Transfer function	Plot	Equation	Range
Log-sigmoid		$f(x) = \frac{1}{1 + e^{-x}}$	$[0, +1]$
Hyperbolic tangent sigmoid		$f(x) = \frac{2}{1 + e^{-2x}} - 1$	$[-1, +1]$
Sinusoid		$f(x) = \sin(x)$	$[-1, +1]$
Hard limit		$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$	$[0, +1]$
Linear		$f(x) = x$	$[-\infty, +\infty]$

### 2.2.4 Backpropagation neural networks

As discussed in the preceding section, a backpropagation neural network (BPNN) consists of three basic layers called the input layer, output layer and a number of sandwiched hidden layers (**Figure 2.7**). The connected layered neurons allow the information received from input neurons to flow in the forward direction only. The strength of the information is manipulated by the weighted synaptic connections between the inter-layer neurons. The weighted signal entering the hidden layer neurons is processed by summing up the signal and subsequently applying an appropriate transfer function. The transfer functions commonly used in hidden layers are generally of continuous and differentiable nature namely, log-sigmoid and tangent hyperbolic functions. The information processed sequentially through a number of



hidden layers and is presented at the output layer of neurons. A comparison of the signal received at the output layer of neurons with the actual or target value allows one to estimate precisely whether the neural network has learned from the training examples or not. A neural network is said to be trained if there is proximity between target values and predicted outputs or the network error has reached a minimum threshold value. It is brought about by a gradient descent algorithm commonly known as a Backpropagation (BP) algorithm. The algorithm adjusts the weights and biases of the neural network by calculating the network error commonly in terms of squared error namely, mean square error (MSE) or the sum of squared error (SSE) and back-propagating this error to the move the weights and biases along the negative of the gradient of computed error.

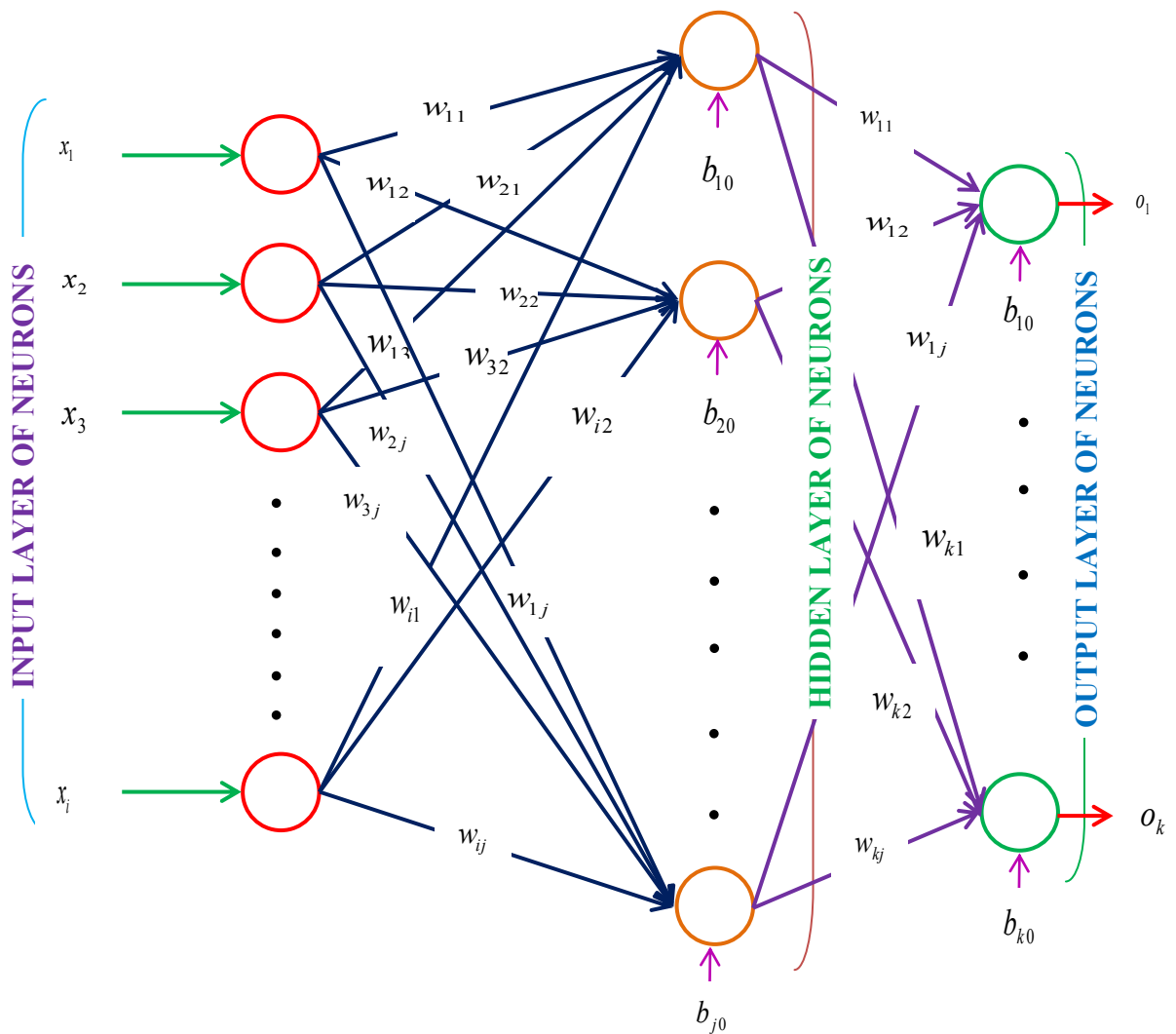


Figure 2.7: Backpropagation neural network

The entire process of this algorithm can be narrated in the following steps:

**Step 1:** Initialization of BP algorithm with random values of weights and biases.

**Step 2:** The neural network having  $i$  input neurons,  $j$  hidden layer neurons, and  $k$  output neurons is presented with information in the form of input-output pairs representing the training pattern  $\zeta$ . The independent variables present in the training pattern represent the input neurons  $x_i$  and the dependent variables are represented by the output neurons  $o_k$ .

**Step 3:** Forward propagation of the information through the hidden layer neurons and computing of output of hidden layer neuron for each training pattern  $\zeta$ . The output of hidden layer neurons  $o_j^\zeta$  is computed as  $o_j^\zeta = f(\text{net}_j^\zeta) = f\left(\sum_i (x_i w_{ij}) + b_{j0}\right)$ , where  $f$  is the transfer function for the hidden layer neurons,  $w_{ij}$  is the strength or magnitude of the connection between input and hidden layer neurons and  $b_{j0}$  is the value of bias attached to the hidden layer neurons.

**Step 4:** Forward propagation of information computed in Step 3 to the output layer and evaluating the output at the output layer of neurons. The computed value of output for the output layer of neurons for the entire training pattern is  $o_k^\zeta = f(\text{net}_k^\zeta) = f\left(\sum_j (o_j w_{jk}) + b_{k0}\right)$ , where  $f$  is the transfer function for the output layer neurons,  $w_{jk}$  is the strength or magnitude of the connection between hidden and the output layer neurons and  $b_{k0}$  is the value of bias attached to the output layer neurons. The output  $o_k$  can also be represented as

$$o_k^\zeta = f\left(\sum_{\zeta} \left( f\left(\sum_i (x_i w_{ij}) + b_{j0}\right) w_{jk} + b_{k0}\right)\right)$$

**Step 5:** Compute the error between the target value and the predicted output

$$E = \frac{1}{2} \sum_{\zeta} (t_k^\zeta - o_k^\zeta)^2 = \frac{1}{2} \sum_{\zeta} \left[ t_k^\zeta - f\left(\sum_j \left( f\left(\sum_i (x_i w_{ij}) + b_{j0}\right) w_{jk} + b_{k0}\right)\right) \right]^2$$

**Step 6:** Apply the steepest descent algorithm to adjust the weights by backpropagation

of the error computed in Step 5. For output neurons  $\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}}$ .

$$\therefore \Delta w_{jk} = \eta \cdot \sum_{\zeta} (t_k^{\zeta} - o_k^{\zeta}) \cdot f'(net_k^{\zeta}) \cdot o_j^{\zeta} = \eta \cdot \sum_{\zeta} \delta_k^{\zeta} \cdot o_j^{\zeta}, \text{ where } \delta_k^{\zeta} = (t_k^{\zeta} - o_k^{\zeta}) \cdot f'(net_k^{\zeta}).$$

For hidden layer neurons applying the above procedure, we get  $\Delta w_{ij} = \eta \cdot \sum_{\zeta} \delta_j^{\zeta} \cdot x_i^{\zeta}$ ,

where  $\delta_j^{\zeta} = f'(net_j^{\zeta}) \cdot \sum_k w_{jk} \cdot \delta_k^{\zeta}$ . Hence, for a synaptic weight connection from

neuron  $p \rightarrow q$  we have a generalized rule for updating the weights  $w_{pq}^{new} = w_{pq}^{old} + \Delta w_{pq}$

, where  $\Delta w_{pq} = \eta \cdot \sum_{\zeta} \delta_q \cdot o_p$  and  $\eta$  is the learning rate. A large value of the learning

rate  $\eta$  leads to faster convergence, but may result in overshooting of optimal values of the weights. The problem is counteracted by introducing a momentum factor  $\alpha$  into the weight updating algorithm. The momentum factor  $\alpha$  by utilizing the effect of previous weight change on the current weight change provides a smoothing effect to weight oscillations rendered by using a higher learning rate. A simplified weight updating relation given by **Erb (1993)** shows the effect of both learning rate  $\eta$  and the momentum factor  $\alpha$  as:

$$\text{Current change in weight} = \text{learning rate} \times (\text{error}) + \text{momentum factor} \times (\text{previous change in weight})$$

The steepest gradient descent principle utilized by a standard backpropagation procedure is a local optimization algorithm that exhibits good convergence when the weights are located in the proximity of a minimum point, but slower convergence when the weights are located far away from the desired minimum. To address this problem **Hagan and Menhaj (1994)** presented the Levenberg-Marquardt algorithm for training the BPNN. The Levenberg-Marquardt backpropagation algorithm can be regarded as a trade-off between the conventional gradient descent and Gauss-Newton method as it utilizes the advantage of fast convergence through non-linear least square optimization rendered by Newton method and the stability provided by gradient descent through maximum neighborhood principle. Although the weight updating using the LM algorithm increases the convergence rate of the backpropagation algorithm, it still carries the drawback of

getting trapped at the local minima. The LM algorithm updates the weights according to the **Equation 2.1**.

$$w_{n+1} = w_n - H_n^{-1} J_n^T e_n \quad (2.1)$$

where,  $H$  is the Hessian matrix given by,  $e$  is the error vector and  $w$  is the weight matrix.

The Newton method requires calculation of the Hessian matrix  $H$  given by  $H_n = \frac{\partial^2 E_n}{\partial w_{n-1}^2}$ , where  $E$  is the error function and  $w$  is the weight matrix. Due to the complexity of the evaluation of Hessian matrix, an approximate method called the quasi-Newton method is employed by the LM algorithm to compute the Hessian matrix  $H_n = J_n^T \cdot J + \mu \cdot I$ , where  $J$  is the Jacobian matrix,  $\mu$  is the Marquardt parameter and  $I$  is the identity matrix. The Jacobian matrix  $J$  is evaluated by computing the first order derivative of the network error given by  $J_n = \frac{\partial e}{\partial w_{n-1}}$ . For large values of the parameter,  $\mu$  the above expression approximates a gradient descent while a small  $\mu$  makes the algorithm behave as a Gauss-Newton algorithm. Therefore, the parameter  $\mu$  controls the transformation of the LM algorithm from Gauss-Newton algorithm to gradient descent algorithm and vice-versa. Since the gradient descent algorithm possesses inherent drawback of slow convergence, the LM algorithm attempts to shift to Gauss-Newton method as quickly as possible near the vicinity of an error minimum to enable accurate and faster convergence (**Samani et al., 2007**).

### 2.3 Genetic Algorithms

Evolutionary Computation (EC) is an optimization paradigm inspired by evolutionary ideas of natural selection and genetics. The algorithms falling under the domain of EC simulate evolutionary processes in a computing environment and possess notable characteristics, namely an iterative procedure with gradual improvement in the quality of solution, work on a population of solutions, randomized but guided search through exploitation of the historical information, parallel

processing of all possible solutions at a time and the most importantly, it is biologically inspired. The evolutionary algorithms, classified into four major streams namely, Genetic Algorithms, Evolution Strategies, Evolutionary Programming and Genetic Programming, attempt to evolve best possible solutions to a problem through survival of the fittest heuristic. The genetic algorithms (GA) belong to a class of adaptive heuristic search and optimization algorithms with its foundation resting on Darwin's *Theory of natural evolution*. In contrast to steepest descent algorithms that perform a deterministic search by exploiting the gradient information of differentiable and continuous objective functions, the GA's gradient free stochastic search employs computational techniques that follow the traits of *survival of the fittest* heuristic to enhance the quality of solution through exploration and exploitation of the search space gradually. The key features that distinguish GA from the conventional optimization algorithms can be summarized as:

- a) GA works on a coded version of the possible solution to a problem and not on the solution itself as exhibited by conventional optimization algorithms.
- b) In contrast to traditional methods of optimization operating on a single solution at a time, the GA starts with multiple solutions in different directions of the search space, reducing the risk of falling into a local minimum, consequently improving the probability of reaching a global optimum.
- c) The conventional methods rely on gradient information regarding the problem to be optimized, applicable only to the continuous nature of the function. The incorporation of fitness function attributes a gradient-free approach to GA to aptly deal with continuous as well as the discrete nature of optimization problems.
- d) In GA, a gradual improvement in the quality of the solution is brought about by stochastic operators while the conventional methods employ deterministic operators.

### **2.3.1 Historical background**

The idea of harnessing the principles of natural evolution and genetics for developing evolutionary systems for engineering optimization problems started in the latter part of 1950's. In 1960, the idea of Evolutionary Computing was put forth by

Ingo Rechenberg in his work *Evolutionary Strategies*. In 1966, Fogel, Owens, and Walsh developed *Evolutionary Programming*. Both the developments used mutation and selection interrelated to Darwin's theory of evolution. The field of Evolutionary Computing was in the nascent phase until in the year 1975; John Holland published his book *Adaptation in Natural and Artificial Systems*. Holland propounded the population based algorithm based crossover, inversion, and mutation. The book was instrumental in creating a flourishing field that is now commonly known as *Genetic Algorithms*. The sustainable development in the area of GA till date is attributed to David Goldberg's book *Genetic Algorithms in Search, Optimization and Machine Learning*. In the year 1992, John Koza introduced *Genetic Programming* that utilizes a set of computer programs to replicate an individual to discover or generate new programs that can solve a particular problem either accurately or approximately. With each passing decade, there has been a tremendous growth in the application of Evolutionary Computing for solving multi-dimensional problems faced with multi-disciplinary fields. The gap between the various methodologies of Evolutionary Computations is narrowing down with more and more algorithms finding their place under this common umbrella of optimization algorithms.

### **2.3.2 Structure and working: A biological perspective**

The human/animal body is composed of core units called the cell resembling small factories working together. Each cell contains a set of chromosomes (23 pairs in human beings) made of DNA representing the genetic information encoded in genes. Each gene carries a particular trait of an individual and maintains a unique location in the chromosome search space known as its locus. A combination of genes addressing a specific property in an individual are called alleles. A gene carrying different alleles constitutes a gene pool that represents all possible variations that an individual can carry for its future generations. The genome represents a set of genes representing a particular species, whereas the combination of genes for an individual is called as genotype. The information carried by the genotype is essential for constructing an organism known as the phenotype.

In an evolutionary process, new chromosomes are formed through crossover or recombination in which genetic information from two parents is combined to render a new chromosome. In biological terms, reproduction is carried out by Mitosis and

Meiosis. The mutation inadvertently occurs during transfer of genes from parent to the children due to an error in the copying of genes or some unknown environmental reasons. The ability of an individual to reproduce is affected by their capacity to adapt to adverse environmental conditions. Nature's extremes inculcate a continuous struggle for existence in an individual, thereby making the fit individuals prevail and carry their genetic traits to their next generation.

Inspired by the biological evolution narrated above, the GA comprises of four distinct components namely, encoding of possible solutions as chromosomes, fitness or objective function, selection, recombination, and evolution. The main strength of GA is attributed to these standard components, as these can be easily implemented for different forms and applications of GA with minor modifications (McCall, 2005). A typical genetic algorithm cycle exhibited in **Figure 2.8** shows that the GA starts with an initial population of chromosomes that contain a number of possible solutions to a problem coded as genes. The set of these possible solutions coded as genes forms a genotype. The genotype carries the necessary information to create a phenotype, which is decoded to represent a possible solution to a problem. The fitness in GA is evaluated by computing the value of the objective function for its phenotype. The objective function evaluates the efficacy of the solution, i.e. the closeness of the solution to the optimal one. The evolution operator namely, selection operator utilizes chromosome's fitness value as a measure to select the chromosome for possible representation in the next generation.

The genetic operators namely, crossover and mutation acts on the selected group of chromosomes and alters its characteristics to generate the offsprings. The crossover operator utilizes the genetic material from the two parent chromosomes in the hope of creating better offsprings. By doing so, the crossover operator helps the GA in narrowing down the search area by exploiting the fitness of the parent chromosomes present in the current population. The use of the only crossover would attract faster convergence and narrowing of search space but in doing so, the GA may overshoot the global optimum. To overcome this problem, mutation is introduced. The mutation operator alters or mutates the current population of chromosomes by adding some extra characteristics in the hope of deriving fitter individuals which might have been inadvertently lost during the process of selection and crossover, thus providing a second chance of enriching the current population with fitter chromosomes. By

performing mutations in the populations, the search space of GA is expanded allowing the GA to search the surroundings for global minima thus mitigating the chance of falling into local minima. This phenomenon renders search to proceed in small steps so that the global minimum is sought in the vicinity of the local minima located in the past. The process of crossover and mutation is repeated until the complete successor population of chromosomes is produced. The fitness of the offsprings are evaluated using the same objective function and based on their fitness, the current population of chromosomes is replaced with the offsprings. This cycle is continued till the desired termination criteria are satisfied.

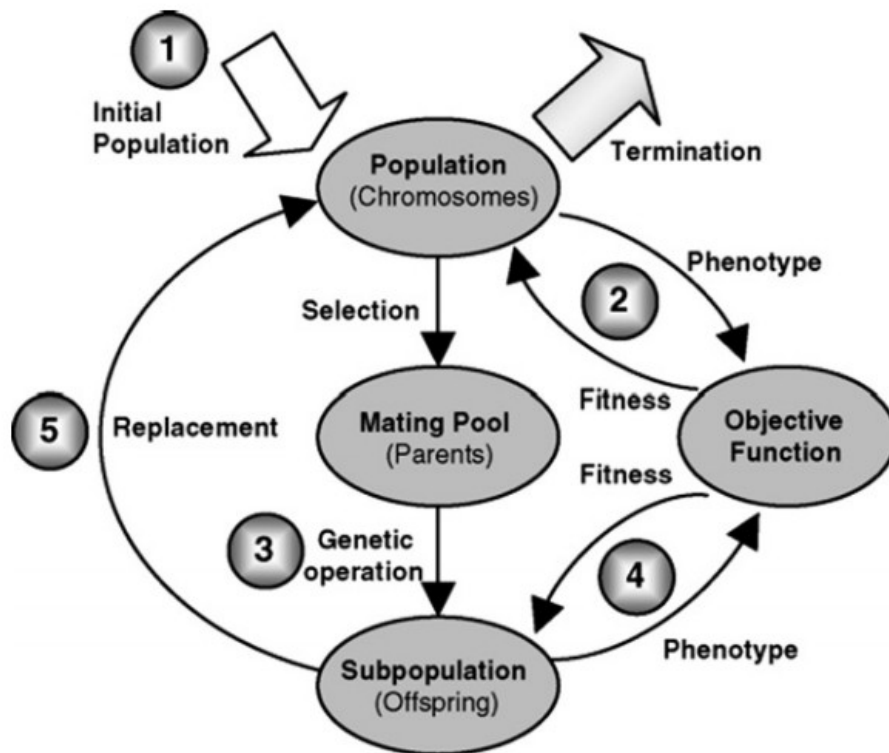


Figure 2.8: Genetic algorithm cycle (Source: Saemi *et al.*, 2007)

### 2.3.3 Operators

#### (i) Selection

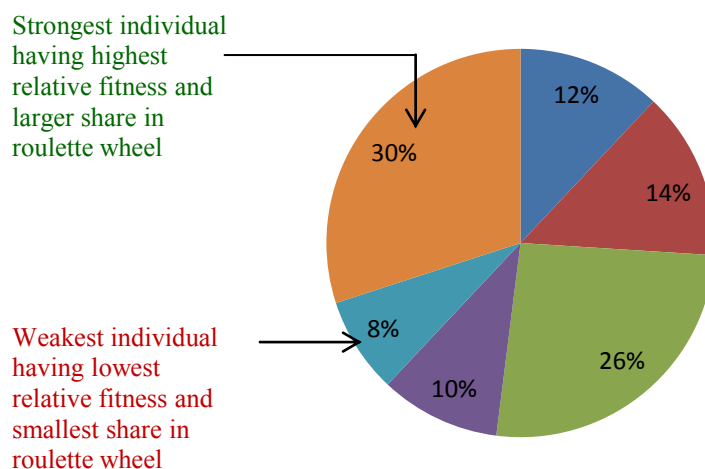
GA is a stochastic population-based search and optimization algorithm which attempts to emulate the biological evolution in a computing environment. The GA, therefore, works on an initial set of solutions using the evolution operator namely,



selection and genetic operators namely, crossover and mutation. The selection operator helps in choosing the parent chromosomes from the population for mating. The selection is performed in a manner, to allow the fitter chromosomes to mate, hoping that they would produce fitter offsprings. To evaluate the quality of chromosomes as a possible solution to the problem, an objective function or fitness function is defined. Based on the fitness, a selective pressure is applied to the chromosome population. The selection operator by doing so directs the algorithm to regions of search space where there is increased probability of finding an optimal solution. Commonly used selection methods are:

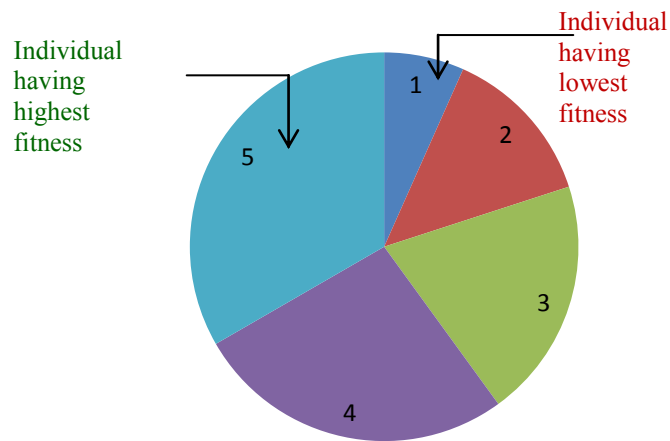
- a) Roulette wheel selection
- b) Rank selection
- c) Tournament selection
- d) Elitist selection
- e) Stochastic universal sampling

The Roulette wheel selection is a traditional selection method employed in GA. In this selection method, the chromosomes are selected based on their relative fitness. A roulette wheel is imagined having a number of pockets or slots filled with chromosomes, whose size depends on the individual chromosome's fitness. For a population having N chromosomes, the wheel is spun N times. The individual on the wheel at which marker stops is selected to form the mating pool of parents. The Roulette wheel selection is shown in **Figure 2.9**.



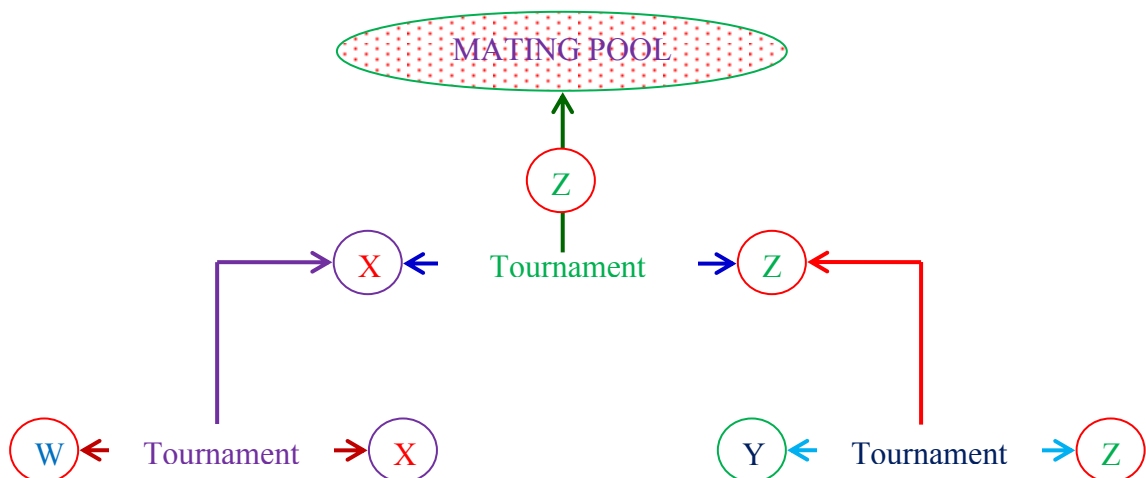
**Figure 2.9:** Roulette wheel selection

The roulette wheel selection may cause some problem if there is a large difference between the fitness of chromosomes, with the best chromosome occupying almost the entire area of the roulette wheel. In such cases, the weaker chromosomes will have a negligible chance to be selected. The rank selection shown in **Figure 2.10**, sorts the chromosomes in the order of their fitness values. The worst chromosome has fitness 1 and best has fitness  $N$ . This selection strategy allows maintaining of diversity within the population but results in slower convergence.



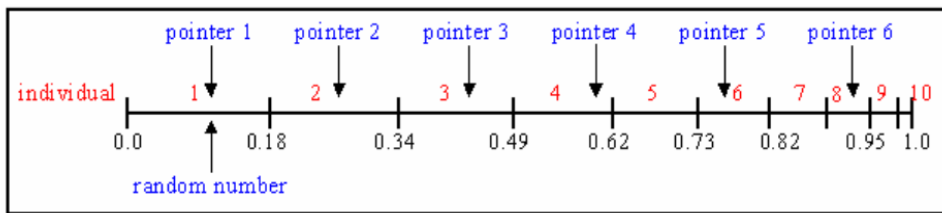
**Figure 2.10:** Rank selection

In a tournament selection shown in **Figure 2.11**,  $N_U$  individuals are selected at a time with uniform probability and are allowed to compete in a tournament. The winner of the competition is the individual with higher fitness and is inserted into the mating pool. This process is repeated till the mating pool is filled with individuals.



**Figure 2.11:** Tournament selection

The elitist selection scheme ensures that the best individual is not lost and continues to prevail in the next generation, thereby maintaining its influence on the rest of the population. This strategy is a safeguard against the inadvertent loss of best individual during the crossover and mutation operations. Using this approach the quality of the solution monotonically increases over generations. In a stochastic universal sampling, the relative fitness of all individuals is evaluated and plotted on a line such that the length of the segment is proportionate to an individual's fitness (Figure 2.11). Equally placed pointers are put on this line. The distance between each pointer is  $\frac{1}{N}$  where  $N$  is the number of individuals to be selected. If six individuals are to be selected, then the pointer distance should be  $1/6$  or  $0.167$ . The random numbers are generated between 0 and 1. The random number will fall on the line close to the position of the pointer. For selecting a single individual, a random number is drawn in the range  $[0, 0.167]$  as shown in **Figure 2.12**. The process is repeated till all the individuals are selected.



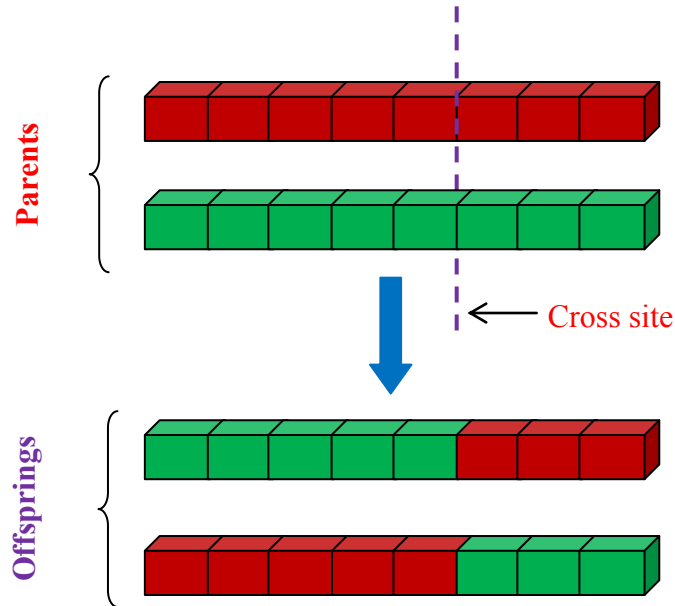
**Figure 2.12:** Stochastic sampling (Source: Pencheva *et al.*, 2009)

**(ii) Crossover**

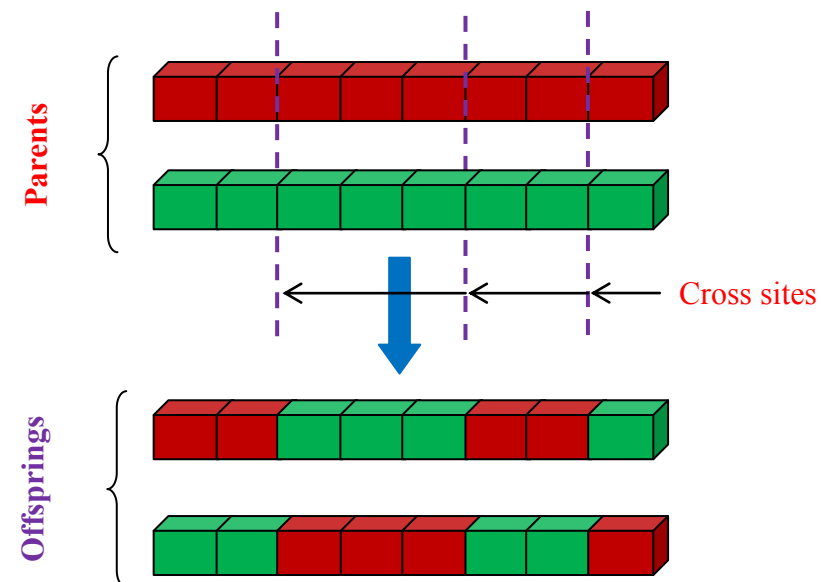
Crossover or recombination operator works on two parent chromosomes to produce an offspring. The crossover operator thus creates new variants of the chromosome population by swapping parts of chromosomes, i.e. genes between two parent chromosomes across randomly selected crossing points (Senouci and Al-Ansari, 2009). Some of the commonly used crossover strategies are:

- a) Single-point crossover
- b) Multi-point crossover
- c) Uniform crossover
- d) Scattered crossover

A single point crossover uses a single cross-site for a pair of parental chromosomes (**Figure 2.13**), whereas a multi-point crossover introduces numerous cut points in the parent chromosomes for swapping of genetic material and ultimately creating two offsprings (**Figure 2.14**).

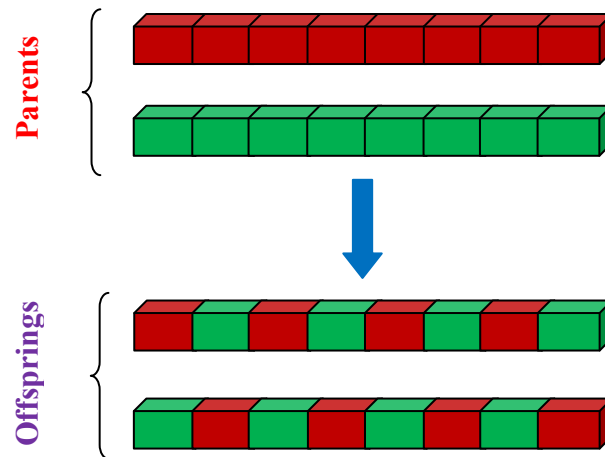


**Figure 2.13:** Single-point crossover



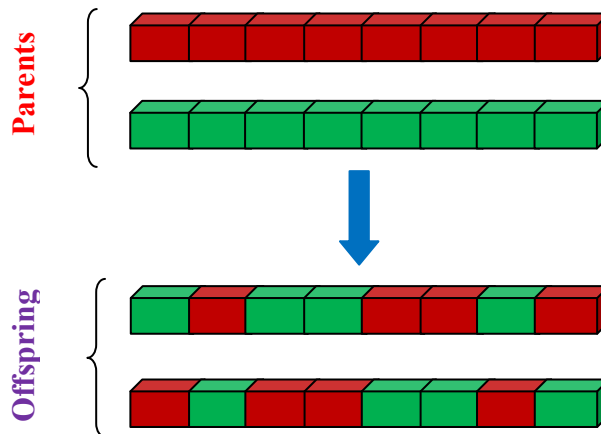
**Figure 2.14:** Multi-point crossover

The uniform crossover operator does not use cross sites; rather it employs a random number generator to indicate the swapping of a particular gene across the parent chromosomes (**Figure 2.15**).



**Figure 2.15:** Uniform crossover

The biggest problem with one point and two point crossover is its dependency on the position of the cross site. Use of these crossover strategies may result in offsprings having almost identical chromosomal configurations in the subsequent generations. Sometimes it may take many generations to evolve a fitter offspring carrying potentially useful characteristics of the parent chromosomes. The scattered crossover strategy removes the dependency on crossover point by randomly selecting some genes from one parent and some genes from the other parent chromosome (**Figure 2.16**).



**Figure 2.16:** Scattered crossover

**(iii) Mutation**

Mutation operator helps in maintaining diversity within the chromosome population by randomly altering the genes of each chromosome (**Chen *et al.*, 2014**) by

adding new characteristics. Applying mutation in GA increases its search space. Mutation can be brought about by:

- a) Flipping
- b) Interchanging
- c) Reversing

In the case of flipping a parent and mutation chromosome is generated randomly. If the bit value 1 occurs in mutation chromosome, the value in parent chromosomes is flipped from 0 to 1 and from 1 to 0 to produce a mutated child (Figure 2.17).

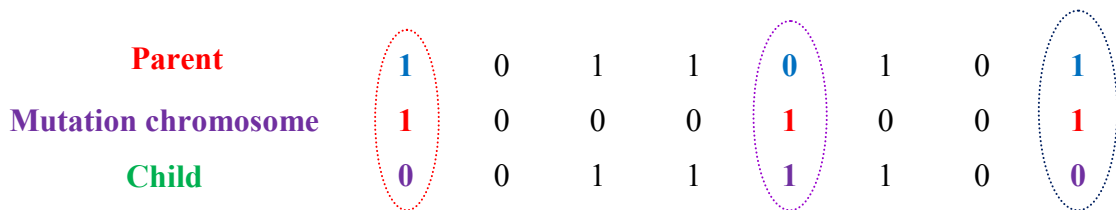


Figure 2.17: Mutation using flipping

In the case of interchanging, two random positions along the chromosome length are chosen and after that, their values are interchanged (Figure 2.18).

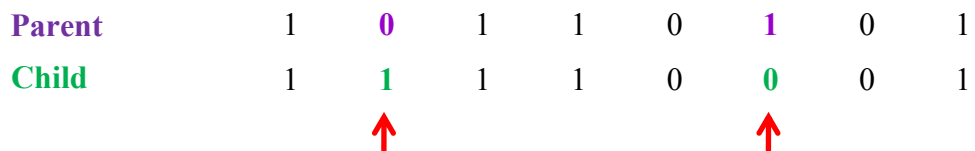


Figure 2.18: Mutation using interchanging

Reversing chooses a random position along the string length and values of genes next to the chosen position are reversed (Figure 2.19).

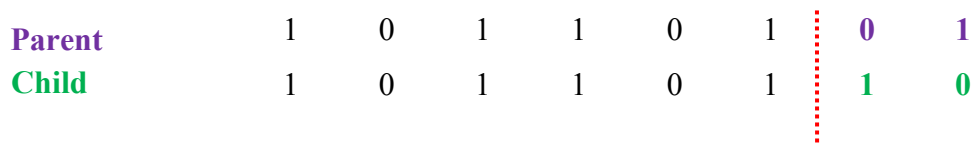


Figure 2.19: Mutation using reversing

### 2.3.4 Parameters

The parameters of GA, namely size of population, crossover rate and mutation rate bear an enormous impact on its search mechanism. The typical value of

parameter settings is given by some researchers (**Grenfensette, 1986; Dejong and Spears, 1990**). However, these cannot be universalized and are problem specific. Brief introduction to the various parameters used in GA are narrated as under:

**(i) Population size**

A larger population size promotes the probability of finding an optimal solution by discriminating between the good and bad building blocks but, may increase the time for GA to converge. However, if the population size is small, then the quality of the solution is left to the vagaries of chance. The typical range of population size is 20 to 200.

**(ii) Crossover rate**

The crossover rate determines how often the bits of strings are to be swapped across a pair of chromosomes. The crossover rate ranges from 0 to 1. A higher value of crossover rate promotes in generating more offsprings by combining the parent chromosomes, helping in the exploitation of the current population of chromosomes.

**(iii) Mutation rate**

The mutation rate determines the probability of genes in chromosomes undergoing alterations. The mutation rate ranges from 0 to 1 and controls the speed of GA in exploring the new areas in the search space. A lower value finds GA stuck in a local minimum whereas a higher value increases the search space to a level, wherein the possibility of GA converging to an optimal solution becomes rare.

## **2.4 Summary**

Artificial Neural Networks (ANN) and Genetic Algorithms (GA) are amongst the most popular and widely used Soft Computing techniques that emulate the biological processes of learning and genetic evolution respectively. Although, the history of neural networks is quite old yet, the credit for making the neural network methodology popular goes to Rumelhart, Hinton and Williams for discovering the backpropagation algorithm in 1986.

The artificial neuron forms the basic computational unit in a neural network. A neural network is characterized by three basic entities namely, the arrangement of neurons and their connections, called its architecture or topology, learning algorithm for updating or adjusting the weights associated with connections and the transfer functions associated with the neuron layers. Amongst the numerous neural network topologies and learning algorithms, the Multilayer Feedforward Neural Network (MFNN) trained using error backpropagation or simply backpropagation (BP) known as Backpropagation Neural Network (BPNN) is the most popular and widely used for applications related to function approximation and pattern recognition. The architecture of BPNN comprises of three basic layers of neurons namely, an input layer, an output layer and a number of intermediate hidden layers. BP algorithm employs the principle of steepest descent to update the weights and biases based on the information presented to the MFNN. The Levenberg-Marquardt (LM) backpropagation algorithm is an improved BP algorithm that utilizes the conventional gradient descent and Gauss-Newton method to provide both efficiency and fast convergence.

The genetic algorithms (GA) are population-based evolutionary algorithms with their foundation resting on Darwin's *Theory of natural evolution*. The Genetic Algorithms came into vogue in the year 1975 when John Holland published his book *Adaptation in Natural and Artificial Systems*. The GA comprises of four distinct components namely, encoding of possible solutions as chromosomes, fitness or objective function, selection, recombination, and evolution. The solutions to the problems are encoded as genes present in the chromosomes. The GA starts with a population of chromosomes and at each generation the fitness of chromosomes is evaluated using a fitness function. The fitter chromosomes are filtered using a selection operator and are allowed to form the new generation of chromosomes. The crossover and mutation operators help the GA to create next generation of the population. The crossover operator exploits the current population whereas the mutation operator helps in maintaining the genetic diversity within the population. The parameters of GA, namely size of population, crossover rate and mutation rate bear an enormous impact on its search mechanism are therefore must be chosen judiciously to strike a balance between the quality of solution and the computational effort involved.



## **Chapter 3**

### **Literature Survey**

---

- **Introduction**
  - **Artificial Neural Networks applications**
  - **Hybrid Genetic Algorithms-Artificial Neural Networks applications**
  - **Conclusions of the literature survey**
  - **Research gap**
-

### **3.1 Introduction**

Neural networks inheriting the simplified biological structure and working of the human brain have been in vogue for the last two decades for modeling unstructured problems associated with multidisciplinary fields of engineering. The closeness to human perception has allowed neural network's amalgamation in areas associated with the prediction of the complex material behavior. Among the numerous neural network architectures, the Multilayer Feedforward Neural Networks (MFNN) have been widely used by the researchers for developing mathematical models for highly complex and non-linear interactions between independent (input) and dependent (output) variables. The architecture of MFNN is represented by a number of artificial neurons arranged in layers that are connected in the forward direction only. The neural network weights form a link between the inter-layer neurons and therefore allow transmission of information from input layer neurons to the output layer neurons. The learning of the MFNN is brought about by error backpropagation algorithm or commonly called the backpropagation (BP) algorithm.

The information presented to the neural network flows from the input layer to the output layer through a series of hidden layers. Based on the nature of information, the BP algorithm systematically updates the neural network weights and biases to reduce the error between the actual output and the output predicted by the neural network predicted to an acceptable minimum. The value of weights and biases used for initializing the BP algorithm, therefore play an important role in the efficient learning of the neural network. The neural network weights and biases located on local grade forces the BP algorithm's entrapment at the local minima, significantly affecting the learning ability of the neural networks and render slow convergence to the global optimum. To cover up the inherent drawback of BP algorithm's local search, the global search ability of Genetic Algorithms (GA) is hybridized with the neural networks during its training phase. The comprehensive literature review presented in the subsequent sections deal with the potential and the versatility of artificial neural networks (ANN) highlighting its applications and its hybridization with GA for modeling problems associated with modeling material properties of concrete. Based on the findings of the literature survey, the research gap has been identified and discussed.

## 3.2 Artificial Neural Networks applications

### 3.2.1 Applications in modeling various properties of concrete

The constituents of concrete namely, cement, fine aggregate, coarse aggregate, admixture, and water, render composite nature to the concrete and makes its material modeling a challenging task. Although, empirical relations in the form of regression equations derived from experimental data are traditionally in use for extracting knowledge about the properties of concrete, yet these do not give the liberty or flexibility for incorporating a number of different factors affecting the properties of concrete. Moreover, the empirical relations do not yield acceptable prediction accuracy wherein there are number variables associated with a complex physical phenomenon. The ability to map such complex interactions among the properties of concrete and its constituents is attributed to ANN's potential to extract knowledge through learning and adaptability to change in the presented information. The literature review elaborates the applications of ANN wherein it has been widely applied for modeling different properties of concrete.

**Hodhod and Ahmed (2014)** employed four layer backpropagation neural network architecture comprising of four input neurons namely, concrete cover depth, coefficient of chloride diffusion, threshold value of chloride and chloride concentration at the surface of concrete containing slag. The output neuron of the model comprised of time required for initiation of corrosion. The effects of the input parameters were studied by comparing the neural network predicted value and those derived from Fick's second law of diffusion. The study showed an increase in time of corrosion initiation with an increase in cover of concrete and the threshold value of chloride. A decrease in chloride initiation time was noticed with a decrease in the chloride concentration at the surface and coefficient of diffusion for chloride ions. A close correlation between the value obtained from the empirical relationship and neural network predicted corrosion initiation time proved that, the neural network can be effectively utilized for learning the complex material behavior of concrete.

**Duan and Poon (2014)** constructed two neural networks based on the data collected from past literature related to recycled aggregate concrete (RAC) for studying the importance of the factors affecting the compressive strength and elastic modulus of RAC. Sixteen variables influencing the properties of RAC were identified

and separated into two parts, one describing the factors that have a direct/predictable impact and the other having an indirect/unpredictable impact. The variables were used as input neurons for the neural network model. The compressive strength and elastic modulus of RAC represented the output neurons for the model. The results showed that the compressive strength and elastic modulus of RAC can be accurately predicted using the neural network methodology. Moreover, by incorporating the factors having an indirect or unpredictable impact into the neural network model, an enhancement in the accuracy of prediction was noticed.

A non-parametric approach incorporating the use of neural networks was harnessed by **Bal and Buyle-Bodin (2014)** for modeling creep of concrete. The data for the study were collectively collected from the past literature and by performing experiments. The model of the neural network comprised of twelve input neurons parameters influencing the creep of concrete and one output neuron representing the creep of concrete. The hidden layers and their neurons were varied for evolving the best neural network architecture catering to the problem in hand. The prediction accuracy of the neural network model was compared with other traditionally used parametric models. The trained neural network model was further utilized to bring forth the effect of each parameter on the creep of concrete. The study showed the effectiveness of neural network approach for modeling creep of concrete and its ability for correct assessment of various parameters influencing the creep of concrete.

In one of the studies undertaken by **Najigivi *et al.* (2013)**, the neural network was used to draw the functional relationships between the ingredients of concrete blended with nano-silica and ash of rice husk, media required for curing of concrete, time required for curing of concrete and permeability properties of the blended concrete. Levenberg-Marquardt (LM) backpropagation algorithm was employed for training the neural network. The neurons in the hidden layer were varied, and it was seen that the prediction accuracy of the neural network in terms of correlation coefficient reached a threshold value for the neural network model with twenty hidden layer neurons. A close correlation between the experimental outputs and neural network predicted values was noticeable with regard to the permeability of blended concrete. The study demonstrated the usefulness of the artificial neural networks in modeling the complex material behavior of concrete and proved to be an effective alternative to the traditionally used empirical relationships.

The temperature of concrete during curing was modeled using artificial neural networks by **Najafi and Ahangari (2013)** and compared the predictability of the neural networks with the linear and non-linear regression models. The experimental data for the study were collected by measuring the concrete temperature using strain gauges. A neural network model was prepared to model the temperature of concrete based on the time, temperature of the environment, water-cement ratio, quantity of aggregates, height, and diameter of the cylindrical concrete specimen. The neural network models were trained and subsequently tested and validated. The results showed that artificial neural networks could accurately predict the curing temperature with correlation coefficient 0.999 in comparison to linear and non-linear regression models with coefficient of correlation values 0.814 and 0.873 respectively. The study proved that, as compared to linear and non-linear regression models, the artificial neural networks are efficient function approximation tools for non-linear and complex mathematical relationships.

**Lee et al. (2012)** studied the effect of concrete constituents and the state of concrete on the thermal conductivity of concrete using artificial neural networks. The neural network model was trained using the data collected from the past studies. The weights of the neural network were updated using LM backpropagation training algorithm. The numbers of hidden layer neurons were evaluated using trial and error procedure. An optimal neural network model with eleven input neurons and one output neuron was developed. The values of the thermal conductivity predicted by neural network model were found to be in close agreement with the measured values. The study proved that the backpropagation neural networks have the potential of accurately predicting the thermal conductivity of concrete.

**Suryadi et al. (2011)** harnessed artificial neural networks for predicting the setting time for self-compacting concrete. The study was performed on the data collected from a ready mix plant and a concrete laboratory. The neural network was modeled with six input neurons comprising of the design proportions of concrete and one output neuron representing the setting time of concrete. The backpropagation neural network architecture and its training parameters were selected by trial and error. The trained neural network was subsequently tested and validated to avoid over-fitting and to test its generalization ability respectively. The neural network predicted the setting time of the concrete was correlated with the setting time obtained from

experimental studies. The study proved that neural networks can effectively model the complex material behavior of concrete.

**Naderpour et al. (2010)** modeled the compressive strength of fiber reinforced polymer concrete using neural networks to develop the design equations and charts. The neural network model comprised of six input neurons and one output neuron. The six input neurons characterized the properties of fiber reinforced polymer and the dimensions of the test specimen. The databases for the compressive strength of fiber reinforced polymer were collected by performing experiments and by incorporating the data available in the available literature. The compressive strength evaluated using the existing empirical relationships were compared with the neural network predicted compressive strength. The results showed that the neural network was able to accurately predict the compressive strength of the fiber reinforced polymer concrete. Moreover, the percentage error between the compressive strength predicted by empirical models and neural network models was found to be within  $\pm 20\%$  range. Based on the compressive strength predictions provided by the trained neural network an equation was developed, which provided close agreement with the experimental results.

The durability of concrete based on the permeability of chloride ions was analyzed by **Parichatprecha and Nimityongskul (2009)**. The data for the analysis containing the quantity of cement, fly ash, silica fume, water, superplasticizer, coarse aggregate, fine aggregate and water-binder ratio were collected experimentally as well as from previous studies. The concrete specimens manufactured from the design mix proportions were subjected to rapid permeability test to evaluate the resistance of the concrete to chloride ion penetration. After 28 days, each face concrete specimens were placed in contact with NaCl and NaOH solution respectively to enable the movement of chloride ions. An electric current was applied and the amount of charge passed for the duration of 6 hours was recorded. The design mix proportion comprising of eight ingredients were used as input neurons and the total charge passed represented the output neuron for the feedforward neural network model. The number of hidden layer neurons and learning rate were evaluated using trial and error procedure. A multiple linear regression model was also developed considering the data used for training the neural networks. The study showed that in comparison to regression models, the

neural network models are efficient in modeling chloride ion permeability based on the design mix proportions of high-performance concrete.

In one of the studies, the mechanical properties of concrete namely, tensile and compressive strength were modeled using neural network by **Subasi (2009)**. The study was conducted by collecting the experimental data for tensile and compressive tests carried out on the concrete in which percentage of cement was partially replaced with fly ash. The experimental test results were recorded after 2, 7 and 28 days. A backpropagation neural network was used for drawing a functional relationship between the three input neurons namely, percentage content of fly ash, age of specimen in days and unit weight and two output neurons representing the mechanical properties of concrete. The results of the neural network model were compared with the first order regression model. The neural network model was shown to provide higher prediction accuracy than the conventional regression models. The study proved that backpropagation neural networks can be employed for the quick and accurate determination of tensile and compressive strength of fly ash concretes.

**Karthikeyan et al. (2008)** developed neural network model for predicting creep and shrinkage in high-performance concrete. Experimental studies were conducted on the test specimens for 500 days and the relative humidity, volume to surface area ratio, compressive strength, time of loading and time of measuring creep and shrinkage were recorded. These parameters were used as input neurons for preparing a backpropagation neural network with creep coefficients and shrinkage strains as outputs. The hidden layer and hidden layer neurons were selected using trial and error process. Bayesian regularization training algorithm was employed for updating of neural network weights. The neural network was trained using experimental data, and its prediction accuracy was compared with commonly used empirical models. The study showed that the creep and shrinkage values predicted by neural network model were in close agreement with those derived from experiments, proving neural network's applicability in modeling time-dependent behavior of high-performance mixes.

**Topcu and Saridemir (2007)** investigated the use of autoclaved and aerated waste aggregates in the concrete on the various properties of concrete namely, unit weight, strength, ultrasonic pulse velocity and modulus of elasticity. A neural network model was created based on the experimental results obtained by performing tests on

45 specimens. The neural network model comprised of seven input neurons representing the mix proportions of the concrete. The model was used to predict unit weight, ultrasonic pulse velocity, compressive strength, and modulus of elasticity for the concrete. The properties of concrete predicted by the trained neural network were found to be in agreement with the experimental values. The study showed that, artificial neural networks can be used for quick and accurate determination of concrete properties and therefore, have the potential of replacing the cumbersome experimental procedures.

**Kewalramani and Gupta (2006)** performed experimental studies on M20 and M30 grades of concrete specimens of different size and shape and then estimated the compressive strength of concrete based on its weight and Ultrasonic Pulse Velocity (UPV) test results using neural networks. A backpropagation neural network model was developed with two input neurons namely, weight and UPV and one output neuron namely, the compressive strength of concrete. The hidden layer neurons for the model were evaluated using trial and error approach. The neural networks were trained with test data of 336 points each for cube-shaped concrete specimens A and B and 96 points each for cylindrically shaped concrete specimens A and B. The prediction accuracy of the trained neural networks was compared with multiple linear regression models. The study proved that, the neural networks have the capability of accurately drawing the functional relationships between the input-output data and can, therefore, be used as an alternative to non-destructive testing for accurate prediction of the compressive strength of concrete.

In one of the studies, **Hola and Schabowicz (2005)** performed non-destructive tests on seven different concretes and used the experimental data for neural network modeling of compressive strength. Among the different neural network training algorithms, the Levenberg-Marquardt training algorithm was selected for the training of the neural network. Seven inputs obtained through non-destructive testing were mapped to the non-destructive compressive strengths. A close correlation between the experimental and non-destructive compressive strengths with low relative errors was noticed. The trained neural network was then harnessed for predicting compressive strength in two buildings based on the composition of the concrete. The relative errors were computed as 3.70% and 4.84% respectively for the two buildings. The study showed that, the neural networks trained on the non-destructive test data



can be utilized to quickly estimate the compressive strength of building structure with reasonable accuracy.

A methodology for predicting compressive strength of concrete using the design mix proportions was presented by **Kim *et al.* (2004)**. A total 98 data for the study were collected from two different ready mix concrete plants and were modeled using backpropagation neural networks. The artificial neural network predicted compressive strength was compared with the experimental test data provided by the ready-mix companies. The average error in the compressive strengths was computed as 3.9%, depicting close association between the predicted and the actual compressive strength values. The study showed that the trained neural network models have the potential of quickly estimating the compressive strength of concrete before its placement at the site.

**Ince (2004)** proposed a neural network based methodology for assessing the fracture parameters of concrete, by modeling the relationship between three material parameters namely, compressive strength, maximum aggregate size and water-cement ratio and stress intensity factor and crack tip displacement. The neural network model was developed using the test data collected from the past literature. The neural network model was compared with the Two-Parameter Model (TPM). The results of the study proved the capability of neural network models to predict the fracture parameters of concrete accurately. The methodology demonstrated the potential for solving complex problems which are otherwise time-consuming. Moreover, it was shown that, in contrast to empirical relationships having fixed input and output variables, the neural networks has a flexible modeling approach in which a number of other factors can be easily incorporated, and their effect on the output can be easily deduced.

The experimental data of concrete compressive strength were utilized by **Lee (2003)**, for developing a neural network model for assessing the strength of concrete at the site for estimating the appropriate time for formwork removal. Five different neural network models were prepared for conducting this study. The first model was utilized for predicting the early strength of concrete achieved within 24 hours. The second and third models were harnessed for assessment of the compressive strength of concrete on the second and third day of concreting respectively. The fourth and fifth models were utilized for assessment of 7 days and

28 days of concreting respectively. These measurements were recorded with respect to the curing temperature and humidity values. The compressive strength predicted by the trained neural network models were compared with an empirical relationship. The results showed that, the compressive strength predicted by the neural network model were in close association with the experimental data. The study highlighted the flexible nature of neural network modeling, through which sufficient number factors influencing the compressive strength of concrete can be incorporated for developing a sufficiently accurate mathematical model.

**Ni and Wang (2000)** utilized the neural networks to model relationship between the eleven parameters that influence the concrete compressive strength and the 28 days compressive strength of concrete. For conducting the study an MFNN was developed and was trained using experimental test data and data collected from a concrete mixing plant. The trained ANN model was seen to predict the compressive strength of concrete with reasonable accuracy. The trained ANN model was further used to simulate the compressive strength with a change in the cement dosage, sand/aggregate ratio and fineness modulus of sand. The study showed that, neural networks can be very handy for quickly predicting the 28 day compressive strength of concrete. The simulation studies also reveal that neural network models are in close agreement with the rules adopted in the mix proportioning of concrete and can, therefore, be useful for practical applications associated with concrete strength prediction.

**Yeh (1998)** utilized the mathematical modeling abilities of artificial neural networks for developing a strength prediction model for high-performance concrete (HPC). The study was conducted to incorporate the effect of the design mix proportions on the compressive strength of concrete. A backpropagation neural network model having eight input parameters was constructed to predict the strength of HPC. About 1000 samples of trial mixes from 17 different sources were collected for performing the study. Based on the prediction accuracy, the ANN model was compared with a regression formula. The trained neural network model was tested with laboratory experimental data. The trained ANN model was shown to have a higher coefficient of determination than the regression model. The study proved that the neural network modeling provides an accurate prediction for complex materials

such as HPC. Moreover, the study demonstrated the potential of neural networks for studying the effect of age and water-binder ratio on the strength of HPC.

A neural network methodology for predicting the strength of concrete was presented by **Lai and Serra (1997)**. The data for the study were collected from a thermal power station construction site. A backpropagation neural network was formulated with eight input neurons representing the mix proportions of concrete and one output neuron representing the mechanical strength of concrete. The numbers of neurons in the hidden layer were determined using trial and error procedure. The study showed that, the mathematical models based on neural network methodology are promising alternative to analytical formulations wherein a large number of variables are involved.

### 3.2.2 Applications in modeling slump of concrete

Concrete is the most preferred and widely used construction material throughout the world. Every construction activity demands that the concrete mix produced or transported at the site should be of high-density, to assure the stability, compatibility, and mobility of the fresh concrete. A fresh concrete mix satisfying the above qualities is said to be workable. There is no direct test for assessing the workability of concrete, however; it is measured quantitatively by measuring the slump of fresh concrete.

Like other properties of concrete, the workability is affected by the proportions of the constituent materials and their individual characteristics. Several literatures have reported on the effect of the constituent materials of concrete on the workability. Water-cement ratio content is shown to be the most important factor governing the workability of concrete. Increasing the water content increases the amount of lubrication and hence improves fluidity of concrete. **Chindaprasirt et al. (2005)** observed an almost linear correlation between water-cement ratio and workability. The aggregate characteristics in terms of maximum aggregate size, aggregate/cement ratio, fine aggregate/coarse aggregate ratio, and aggregate shape and texture affect the amount of paste required to produce a workable mix and therefore influence the workability of concrete (**Li, 2011**). **Bostanci et al. (2016)** concluded that recycled aggregate concrete mixes required higher superplasticizer demand in order to achieve the desired slump compared to natural

aggregate concrete mixes that could be attributed to the higher water absorption of recycled aggregate.

**Marar and Eren (2011)** reported that increasing the amount of cement in the mixes and decreasing aggregate content leads to an excess of water in the medium and hence, leads to an increase in the workability. However, too much cement content makes concrete sticky and difficult to finish (**Daniel, 2006**). **Pofale and Quadri (2013)** studied the utilization of crusher dust in concrete and found that the manufactured sand particles due to their angular shape and rough surface texture improve the internal friction in the mix thereby, reducing the workability. The natural river sand particles on the other hand owing to their cubical or rounded shape with a smooth surface texture, ensures a good workability of concrete (**Mailar et al., 2016**).

**Muhit (2013)** showed that addition of superplasticizer increase the workability of concrete but their dosage must be limited in the range 0.6%-1%. **Dumne (2014)** showed that that, the concrete containing fly ash and superplasticizer yields good workable mix in addition to a marginal increase in compressive strength. **Jianyong and Yan (2001)** stated that concrete with 30% ground granulated blast-furnace slag replacement level and the same superplasticizer content increased slump value slightly than Portland cement concrete. **Sabet et al. (2013)** reported that “ball-bearing effect” of fly ash concrete with fly ash contents of 10% and 20% increased concrete slump and therefore reduced the amount of superplasticizer required to reach target slump.

The mathematical modeling of concrete slump is thus difficult owing to the different nature, type, and properties of the constituents used in the concrete design mix. In such cases the conventional regression equations do not provide the expected predictability and reliability. The lack of standard empirical relationships to judge the slump of concrete based on its constituents has created the interest of the researchers towards soft computing tools. Artificial Neural Networks (ANN) inspired by the learning mechanism of the human brain, present a simplified approach for modeling unstructured material behavior problems based on the experimental or historical data. The following paragraphs reviews the studies that utilize the complex function approximating ability of ANN for modeling the slump of concrete.

**Bilgil (2012)** studied the rheological properties of fresh concrete by estimating the slump value and Bingham parameters (Yield stress and Viscosity) for High Performance Concrete (HPC) using ANN. The ANN model comprised of six input neurons namely, gravel, sand, fine sand, cement, water and superplasticizer. The slump value, yield and viscosity were considered as the three output neurons. The data of experimental studied were used for training the neural network. The neural network model was able to predict the slump and Bingham parameters close to the experimental values, proving that ANN is viable methodology for determining the rheological characteristics of fresh concrete.

**Boukhatem et al. (2012)** utilized principal component analysis (PCA) along with neural networks for developing a mathematical model for predicting the slump and compressive strength of concrete. The study was performed by collecting data from the past literature and by performing the laboratory tests. The number of factors influencing the slump and the compressive strength of concrete was minimized using PCA. Backpropagation neural network trained using Bayesian Regularization was developed by incorporating the most important factors affecting the slump and compressive strength of concrete. In all six neural network models were developed for studying the inter-relationships among the input parameters and the properties of concrete containing mineral admixtures. The study showed that amalgamation of PCA with neural networks helps in improving the prediction accuracy of the neural network and reduction in the training time taken by the neural network.

**Chine et al. (2010)** used the ANN to model slump of high-performance concrete (HPC). Eleven input parameters influencing the slump of HPC were identified. A multilayer backpropagation neural network was developed using proportions of seven ingredients used to produce HPC in  $\text{kg/m}^3$  namely cement, fly ash, blast furnace slag, water, superplasticizer, coarse aggregate and fine aggregates. In addition to this, four ratios namely, water to cement ratio, water to binder ratio, water to solid ratio and total aggregate to binder ratio were utilized to develop the model. Experimental data from past studies conducted were collected for developing the ANN and multiple regression model. The performance metrics namely, root mean square error (RMSE) and coefficient of determination ( $R^2$ ) was used for assessing the prediction accuracy of the mathematical models. The results showed that trained ANN yielded a lower RMSE and higher  $R^2$  values in comparison to the regression model.

The study proved that ANN methodology can be reliably used to develop mathematical model for nonlinear problems associated with material behavior of concrete.

In one of the studies, **Jain *et al.* (2008)** used 47 laboratory observations comprising of concrete mix constituents and corresponding slump values for developing ANN based mathematical model for estimation of concrete slump. For conducting the study, three neural networks and three multiple linear and non-linear regression models were developed. The first model had mortar and coarse aggregate as input, while the second model had paste sand and coarse aggregates as the input variables. The third model had water, cement, coarse aggregate and sand as input. For all the models, the slump was chosen as the output variable. A multilayer feedforward neural network (MFNN) trained using backpropagation algorithm was employed for conducting the study. A trial and error procedure was employed for determining the optimal number of neurons for the hidden layer. The accuracy of the neural network prediction was compared with regression models using different performance metrics. The results showed that ANN is efficient in modeling the non-linear and complex interactions between the concrete design mix proportions and the concrete slump. The sensitivity analysis conducted in the study showed the effect of mortar, coarse aggregate, paste and sand on the slump of concrete and thus provided an insight into the complex nature of concrete. The slump of concrete was shown to decrease and then rise as the amount of mortar or coarse aggregates was increased. However, the concrete slump increases with paste content and decreases with sand content. The sensitivity analysis thus showed that, there exist a critical level of concrete mix constituent beyond which the behavior of concrete slump changes.

**Yeh (2008)** attempted to model the effects of superplasticizer (SP) and fly ash (FA) on the slump of high performance concrete (HPC). The experimental data for 103 design mix were collected for the study. An ANN model was developed using content of cement, fly ash, blast furnace slag, water, superplasticizer, coarse aggregate and fine aggregate as seven inputs for predicting slump of concrete. A trial and error approach was used to determine the optimal number of hidden neurons. The ANN model was trained using backpropagation algorithm. The performance of ANN model compared with polynomial regression model using RMSE and  $R^2$  statistical metrics demonstrated its potential for accurately predicting slump of concrete. The trained

model was utilized to explore the slump behavior with water at different water-binder ratio, SP-binder ratio and fa-binder ratio. It was shown that partial replacement of cement with fly ash raised the upper limit of the slump. The SP content was shown to augment the slump values without increasing the water content.

An artificial neural network-based mathematical modeling for high-performance concrete slump (HPC) was presented by **Yeh (2007)**. The data comprising of 78 concrete mix proportions and corresponding slump were collected for building the mathematical model. The neural network model comprised of seven neurons representing the proportions of concrete and one output neuron representing the slump value. The prediction accuracy of ANN model was compared with second order regression models. The ANN model showed a close association with the experimental data. The neural network models and regression models were further utilized for developing trace plots indicating the variation of fly ash, water content and superplasticizer on the slump value of concrete. The response trace plots showed a sharp increase and decrease in slump values with fly ash content. The plots of water and superplasticizer were shown to be identical, attaining a saturation level beyond a certain water content and superplasticizer dosage. The study proved the applicability of neural network modeling for concrete slump and its accurate response to the variation of fly ash, water content, and superplasticizer content.

Modeling slump of highly complex material Fly Ash and Slag Concrete (FSC) using an artificial neural network was presented by **Yeh (2006)**. A neural network model with seven input neurons representing the mix proportions of concrete and one output neuron signifying the slump value of concrete with seven hidden layer neurons was created for conducting the study. A total of 78 mix proportions and their corresponding slump values were collected and were used to develop the workability model. The ANN modeling approach was compared with the conventional second order regression model. The results showed that the ANN model was more accurate in predicting the concrete slump. The response plot showed the positive influence of cement, fly ash and slag content on the slump value of concrete. Identical plots for water content and superplasticizer dosage were obtained, exhibiting their threshold limits for maximum slump value. The study showed that the neural network modeling approach can be conveniently used for prediction of concrete slump for any concrete design mix proportion as long as admixtures are of the same properties.

**Oztas et al. (2006)** showed the potential of Artificial Neural Networks (ANN) to predict High Strength Concrete (HSC) slump. 187 design mix samples of HSC were collected from past literature having a compressive strength in the range 40 MPa to 120 MPa. The neural network was modeled with seven input neurons namely, water/binder ratio, fine aggregate ratio, and content of water content, air entraining agent, super-plasticizer, fly ash and silica fume in  $\text{kg/m}^3$ . The output neuron of the model comprised of one neuron representing the slump value of concrete. The collected data was randomized, and 169 sample data were used for neural network training. The remaining 18 samples were employed for testing the reliability of the trained model. The performance of the neural network was tested using four statistical parameters namely, root mean squared error (RMSE), the coefficient of determination ( $R^2$ ), mean absolute percentage error (MAPE) and the sum of squared errors (SSE). A trial and error procedure was adopted for determining the optimal hidden layers and hidden layer neurons. The results of the study showed that the values of artificial neural network predicted slump were in close association with the actual data. The study proved the effectiveness of the neural network to model the complex material behavior of HSC and can, therefore, be used as an alternative to trial and error procedure of concrete mix design.

A mathematical model for predicting workability of concrete containing metakaolin (MK) and fly ash (FA) using artificial neural network was presented by **Bai et al. (2003)**. The data of three standard workability tests namely, slump, compaction factor and Vee Bee time were collected for different concrete proportions and the water-binder ratio (0.4 and 0.5). A partial replacement of cement by pozzolanic materials namely, metakaolin and fly ash was done up to 15% and 40% respectively. Three independent mathematical models were constructed using neural networks methodology for modeling the workability of concrete based on slump test, Vee Bee test, and compaction factor. The results showed that neural network prediction was in close agreement with the observed values of workability. The ANN model was further used to analyze the effect of percentage replacement of cement by metakaolin and fly ash. The slump was shown to decrease with increase in MK replacement indicating its higher surface area that demands more water for desired workability. The fly ash on the other hand is shown to increase workability. However, for mixtures having both MK and fly ash content, the slump value is shown to



decrease. The studies led to the conclusions that ANN based mathematical models trained using experimental data provide an accurate prediction of the concrete slump and can, therefore, be utilized for predicting of slump with a high degree of accuracy for concretes blended with metakaolin and fly ash.

**Dias and Pooliyadda (2001)** used the data of ready mix concretes obtained from three batching plants for modeling the slump and 28-day strength using artificial neural networks. The training dataset comprised of 93 records and the remaining 44 records were utilized for testing of the trained neural network. The mix proportion data was transformed into non-dimensional ratios. Similarly, data from past literature for high strength concrete (HSC) were collected and converted into the non-dimensional ratio. The performance of the neural network for raw data was compared with that trained using dimensionless ratios. The neural networks trained using raw data gave greater prediction accuracy and can be used as a substitute to multiple regression models. The study showed that in comparison to slump modeling, the strength modeling required more input parameters. Moreover, a sensitivity analysis using trained neural network, showed its effectiveness in picking up the primary and secondary parameters affecting the strength of concrete.

### **3.3 Hybrid Genetic Algorithms-Artificial Neural Networks applications**

Nature inspired computational techniques namely, GA and ANN have been hybridized in a number of applications. The complementing nature of these two distinct soft computing approaches has given an impetus to the mathematical modeling paradigm, through the amalgamation of global and local search algorithms. The stochastic global gradient free search of GA has been harnessed for evolving either the architecture of neural networks that included determination of optimal hidden layers and hidden layer neurons, transfer functions and training parameters or optimizing the neural network architecture through the evolution of an optimal set of initial weights and biases. The literature review discussed in the subsequent sections gives a comprehensive overview of the hybrid GA-ANN applications in modeling material behavior of concrete, Civil Engineering and multidisciplinary fields respectively.

### 3.3.1 Applications in modeling material behavior of concrete

**Yu et al. (2016)** presented a knowledge management (KM) methodology for ready-mixed concrete using artificial neural networks coupled with genetic algorithms to facilitate effective production processing. The methodology was applied in the KM system to predict the 28-day concrete compressive strength. The neural network was modeled using water, cement, metakaolin, fine aggregate and coarse aggregate content as input neurons and compressive strength at 28 days as output neuron. The genetic algorithm (GA) is applied to compute the weight and threshold values. The results showed that, in comparison to randomly initialized backpropagation neural network, the neural network initialized with GA optimized weights and biases have improved the convergence rate and prediction accuracy.

The compressive strength of concrete was predicted by **Nikoo et al. (2015)** by using artificial neural networks and genetic algorithms. Cylindrical concrete samples were tested for 28 days compressive strength and the experimental results were utilized for modeling the compressive strength. The water-cement ratio, maximum sand size, amount of gravel, cement, 3/4 sand, 3/8 sand and coefficient of soft sand were used as inputs for ANN model. The number of hidden layers, number of neurons and synaptic weights for the ANN model were optimized using GA. Various learning algorithms were evaluated and compared with EANN for complexity and accuracy. The prediction results were compared with conventional multiple linear regression model (MLR). The evolutionary artificial neural network (EANN) model was shown to outperform the MLR model. The sensitivity results showed that water-cement ratio and 3/8 sand were the most and the least effective parameters respectively for the compressive strength of concrete.

**Yuan et al. (2014)** applied two hybrid models namely, backpropagation neural networks optimized by GA and adaptive-network-based fuzzy inference system (ANIFS) respectively, for investigating the effect of various structured and unstructured factors affecting the concrete compressive strength. For the analysis purpose, 180 sample data comprising of 28-day compressive strength were collected. The neural network model consisted of seven input neurons namely, amount of cement, blast furnace slag, fly ash, water, super-plasticizer, coarse aggregate and fine aggregate. The output neuron comprised of the 28-day compressive strength of concrete. The optimal numbers of hidden layer neuron were evolved using trial and

error procedure. The validation data set comprising of 30 test sample data was used to monitor the trained neural network's generalization ability. GA was used to optimize neural network by evolving the optimal values of initial thresholds and weights. The ANIFS was also applied to the data, for prediction of the compressive strength. The study showed that both hybrid models can be effectively applied to problems, for which linear and non-linear regression models do not provide sufficient accuracy and predictability. A comparison of hybrid GA-ANN and ANFIS showed that ANFIS has ease of use and offers a reliable model for prediction of concrete compressive strength.

**Gopala Krishna Sastry *et al.* (2014)** studied the strength properties of concrete reinforced with steel fibre and harnessed the neural network's data processing ability for the development of a macro-mechanical model. Mechanical properties of concrete namely, compressive strength, split tensile strength, flexural strength, and compaction factor were evaluated experimentally using different water-cement ratio, aggregate-cement ratio, fibre percentage and aspect ratio of steel fibres. An ANN model was developed to predict the mechanical properties based on different water-cement ratio, aggregate-cement ratio, fibre percentage and aspect ratio of steel fibres. The number of hidden layer neurons was determined using trial and error method. A genetic algorithm was hybridized with neural network for evolving the initial weights for backpropagation training algorithm to avoid the drawback of backpropagation algorithm getting trapped at local minima. The study showed that the neural network hybridized with genetic algorithms was able to predict the mechanical properties close to experimental values with percentage accurate close to 95%. The hybridization of a neural network with genetic algorithm thus ensured that the backpropagation algorithm does not fall into local minima, thereby improving prediction accuracy.

**Gorphade *et al.* (2013)** created a prediction model using neural network for three different strength characteristics of High Performance Concrete (HPC) namely, compressive, tensile, flexural and its modulus of elasticity based on four input variables namely, ratio of water-binder and aggregate-binder, type of admixture and replacement of cement by the admixture in percentage. The genetic algorithm was incorporated in the training phase of the neural network to cover up the drawback of backpropagation training algorithm slow convergence to the global minimum. The

optimal initial weights for backpropagation algorithm were evolved using the evolutionary search of genetic algorithms, to aid faster convergence. The neural network model with 24 hidden layer neurons was used for the present study. A total of 300 exemplar patterns were used for modeling the strength of high-performance concrete. The hybrid neural network model took just 2000 training cycles to attain the desired performance goal. The hybrid genetic algorithm based neural network was shown to give prediction accuracy close to 95% accuracy when presented with unseen data. The study proved that the hybrid methodology can be successfully employed for evolving a mathematical model for predicting strength characteristics of HPC.

### 3.3.2 Civil Engineering applications

**Bagheri *et al.* (2015)** hybridized artificial neural networks with genetic algorithms for modeling and optimization of activated sludge bulking for a real wastewater treatment plant. The sludge volume index (SVI) was predicted by hybridizing multi-layer perceptron (MLPANN) and radial basis function artificial neural networks (RBFANN) with genetic algorithm (GA). The weights and biases for the neural network were optimized using GA. The MLPANN was trained using various learning algorithms. The Levenberg-Marquardt (LM) training algorithm provided the ideal model with the lowest RMSE. The results showed that the MLPANN-GA was more efficient than the RBFANN-GA in modeling the SVI. Moreover, an increase in prediction accuracy of all models was noticed when these models were hybridized with GA.

**Momeni *et al.* (2014)** hybridized GA with ANN for modeling the bearing capacity of piles. Dynamic load tests conducted on precast concrete piles were collected and used for modeling the pile bearing capacity. The entire dataset was randomized and divided into two parts namely, training dataset and testing dataset respectively. The neural network architecture comprised of five input neurons namely, cross-sectional area (A), length (L), pile set (S), hammer weight (W) and drop weight (H). The pile bearing capacity in kilo-Newtons was considered as the output neuron. A trial and error procedure was utilized for determining the numbers of hidden layer neurons. The neural network model having eight neurons in the hidden layer gave the minimum testing error and was chosen for the modeling purpose. GA was harnessed for evolving the initial weights and biases of the neural network. The ANN model was

trained using LM training algorithm, because of its efficiency compared to the other backpropagation techniques. The prediction accuracy of hybrid GA-ANN trained using LM algorithm was compared with the conventional ANN using the coefficient of determination ( $R^2$ ) statistical metric. The ANN-GA model gave a higher  $R^2$  value than the conventional ANN model. Enhancement of the prediction accuracy of the ANN model was noticed when the ANN model hybridized with GA. Sensitivity analysis for determining the input parameter influencing the most on the output parameter was also conducted. The sensitivity analysis showed that weight of the hammer and geometrical properties of the pile are the most important parameters influencing the pile bearing capacity.

**Chandre Gowda and Mayya (2014)** developed two models for prediction of streamflow in natural rivers. Both the mathematical models were prepared using neural networks, but with two distinct training algorithms, namely backpropagation algorithm and genetic algorithm. For developing the neural network model, the rainfall data of twelve rain gauge stations and the stream flow data were collected. The rainfall for the current time and lagged by one day and two days along with stream flow for lagged by one day and two days, formed the five input neurons for the neural network. The neural network model had only one output neuron comprising of discharge for the current time. The hidden layers neurons were selected by adopting trial and error procedure. The operators of genetic algorithms namely, selection, crossover were applied for optimizing the weights of the neural network and were used for initializing the training of the neural network using the backpropagation algorithm. The performances of neural network models trained using two distinct methodologies namely; backpropagation and genetic algorithm were compared using five different statistical parameters. The results showed that the neural network model trained using optimized weights derived from genetic algorithms was able to give higher prediction accuracy than the conventional backpropagation neural network. The observed values of the stream flows were found to be in close agreement with those predicted using the genetic algorithm neural network. The study showed that the presented methodology compared to the conventional backpropagation training was effective in dealing with the complexities of hydrological forecasting.

A model for prediction of river water quality was prepared by **Ding et al. (2014)**, by incorporating a hybrid methodology harnessing the strengths of

principal component analysis (PCA), GA and ANN. The water quality data comprising of 23 influencing factors were collected from a lake located in China. The PCA was used to select the most relevant parameters influencing the water quality. Out of 23 water quality parameters, 15 have been chosen using PCA. A total of 2680 sample data were collected and were divided into two categories, namely non-polluted and polluted water. GA was then used to evolve the weights and thresholds for the neural network. The BPNN was trained using LM algorithm to improve the convergence speed. The hybrid model was tested using five-fold cross-validation technique to ascertain its prediction accuracy. The study showed that, in comparison to backpropagation neural network, the ANN model trained using GA was able to reach a smaller mean square error. The study thus proved the efficacy of global search genetic algorithm in minimizing the probability of backpropagation algorithm convergence to a local optimum, thereby improving the prediction accuracy of the ANN.

A model for predicting reference evapotranspiration ( $ET_o$ ) for Mosul station located in Iraq was presented by **Abdullah *et al.* (2014)**. The data comprising of five parameters, namely maximum and minimum air temperature, hours of radiation, relative humidity and wind speed were collected for 26 years. A feedforward backpropagation neural network with the above input parameters and reference evapotranspiration as output parameter was constructed with ten hidden layer neurons. A hybrid genetic algorithm–neural network model was also formulated using the same data, comprising of three hidden layers instead of one used in the feedforward backpropagation neural network. The prediction accuracy of the feedforward backpropagation neural network was first compared with empirical relationships proposed by Penman-Monteith. The study showed that ANN model and hybrid ANN model were able to provide good prediction efficiency and were able to cover up the need for comprehensive data required for prediction of  $ET_o$  using the Penman-Monteith equation. In contrast to the neural network model, the hybrid neural network model showed higher prediction accuracy, thereby providing a suitable substitute for conventional Penman-Monteith equation for prediction of reference evapotranspiration ( $ET_o$ ).

In one of the studies by **Asadi *et al.* (2013)**, a hybrid neural network was employed to study the complex rainfall-runoff interactions. The methodology

comprised of four stages, namely pre-processing of data, application of genetic algorithm (GA) for optimal evolving of the neural network weights, tuning of the optimal weights using Levenberg Marquardt backpropagation (LMBP) algorithm and finally a comparison of the results with the actual runoff data. The pre-processing of data was performed by selecting the variables using stepwise regression and applying the K-means clustering to reduce the complexity of the entire data. The time series data for the last 12 years were collected and using the data pre-processing technique, the rainfall and runoff time series were found to be more related to the runoff. The pre-processed evolutionary Levenberg-Marquardt neural networks (PELMNN) methodology presented in the paper showed faster training, good ability to imbibe complex rainfall-runoff processes and higher accuracy in comparison to the studies performed earlier.

**Miao *et al.* (2013)** collected the concrete dam deflection data for 24 months and analyzed them for prediction of dam deformation. A neural network was modeled with seven neurons in the input layer and one neuron in the output layer. The neurons in the hidden layer were determined using trial and error approach. 20 data samples were used for training, and the rest was used for testing of the trained neural network. Three different neural network models were trained and tested for this purpose, namely a backpropagation neural network using gradient descent to update weights, a Levenberg-Marquardt utilizing strengths of Gauss-Newton algorithm and gradient descent algorithm and a hybrid genetic algorithm trained Levenberg-Marquardt neural network. In the study, the weights and bias of the neural network were evolved using the stochastic search rendered by genetic algorithm. The performances of trained models were compared using the mean square error criterion. The results showed that the neural network model hybridized with genetic algorithm amalgamated the strengths of genetic algorithms and Levenberg-Marquardt algorithm, to give close prediction agreement with the actual data even in the case of scarce training data. The study thus proved the utility of neural network modeling in predicting dam deformation and the ability of genetic algorithm in the improvement of prediction accuracy and faster convergence of the neural network.

A displacement analysis of concrete framed building subjected to earthquake forces was performed by **Nikoo *et al.* (2012)** using genetic algorithm and neural networks. A typical reinforced concrete building having four stories and four bays

were analyzed using a commercial software using acceleration values 0.1 g to 1.5 g. A neural network is created to model the damage caused due to earthquake forces by considering seven inputs namely frequency, acceleration, acceleration 0.1, acceleration 0.2, Peak ground acceleration (PGA), normalized distribution of displacement of frames and time duration of the earthquake. The damage to the building was measured in terms of a damage indicator, and it formed the output neuron for the neural network model. A total of 416 exemplar patterns were generated and trained using three different neural networks, namely Multilayer Perceptron (MLP), Feedforward neural network (FF) and Radial Basis Function neural network (RBF). The architecture of these neural networks was optimized by harnessing genetic algorithm for evolving the number of hidden layers, hidden neurons, activation function and the training algorithm. The prediction accuracy of the neural network models was evaluated by comparing the root mean squared error (RMSE) and correlation coefficient (R). Based on the results, the MLP model in comparison to FF and RBF models was shown to be efficient in predicting structural damage. The study thus proved the effectiveness of evolutionary neural networks in determining damage vulnerability of concrete buildings during earthquakes and can, therefore, assist in decision making for planning alternative retrofitting measures.

**Johari *et al.* (2011)** modeled the mechanical behavior of unsaturated soils by hybridizing neural networks with genetic algorithms. The neural network architecture comprised of eight input neurons namely, initial gravimetric water content, degree of saturation, initial dry density, net mean stress with respect to pore-air pressure, axial strain, deviatoric stress, volumetric strain and soil suction. The three output neurons represented the deviatoric stress, volumetric strain, and suction at the end of each increment of axial strain. The neurons in the hidden layer were deduced using trial and error procedure. The database for neural network training and testing consisted of experimental data obtained from testing 23 unsaturated specimens. Out of these 23 specimens, 18 were prepared by static compression and the rest were prepared by dynamic compression. The 15 test results out of 18 test results obtained from testing of 18 specimens were utilized for training of the network, and the remaining 3 test results were utilized for testing of the trained neural network. Genetic algorithm was used for determination of optimum weights of the backpropagation neural network. The results showed that the hybrid genetic algorithm backpropagation neural network



(GABNN) provided good accuracy in training and testing phases. Sensitivity analysis was also performed to assess the effect of various parameters namely, dry density, the degree of saturation and initial net mean stress and it was found that the trained model was able to correctly capture the material behavior of unsaturated soils.

In one of the studies, **Huang and Wang (2011)** harnessed hybrid neural networks for the time series forecasting. The genetic algorithm stochastic search was applied for finding time-lag for time series, hidden layer neurons, and weights of the neural network. For forecasting purposes, backpropagation algorithm was used. The observed data comprised of rainfall and corresponding monthly stream flow for Liujiang River in China. The conventional and most popular technique of generating a time series namely, auto regressive integrated moving average (ARIMA) was compared with the single neural network, radial basis function neural network and hybrid artificial neural network-genetic algorithm. The models were compared on the basis of three statistical metrics namely, the normalized mean squared error (NMSE), Pearson relative coefficient (PRC) and mean absolute percentage error (MAPE). The values of the performance metrics showed that the neural network model amalgamated with genetic algorithm, was able to provide better rainfall forecasting. The study proved the effectiveness of hybrid neural network for modeling complex rainfall-runoff phenomenon.

**Jalalkamali and Jalalkamali (2011)** used artificial neural networks along with genetic algorithms for creating a mathematical model for estimation of groundwater levels in Kerman province of Iran based on the data of rainfall depth, temperature and depth of water in wells recorded in the last 22 years. In the study, two different hybrid mathematical models, namely Feedforward Neural Network-Genetic Algorithm (FNN-GA) and Recurrent Neural Network-Genetic Algorithm (RNN-GA) were used. Both these models were optimized using genetic algorithms. The genetic algorithms stochastic search was applied for evolving the hidden layer neurons for FNN and RNN models. A number of trials were conducted by altering the input layer neurons. The results showed that an increase in the number of input variables or neurons does not increase the efficiency of prediction; rather it leads to increase in the complexity of the network. Based on the results, the study recommended the ratio of input neurons to hidden layer neurons as 3. The study showed that the hybrid neural network models, FNN-GA, and RNN-GA were able to closely forecast the monthly

water level in the wells although FNN-GA model showed slightly higher prediction accuracy.

A methodology for predicting uplift pressures for Diversion Dam was presented by **Baghalian and Nazari (2011)** by utilizing neural network's function approximation ability and genetic algorithm stochastic search for determining the set of optimal initial weights. A multilayer feedforward neural network (MFNN) was used for modeling uplift pressure based on the piezometric head measured with respect to X and Y coordinates of the dam, computed by solving the Laplace's Equation. The neural network training was performed by utilizing two different techniques namely, genetic algorithms and backpropagation. The genetic algorithms trained neural network provided a lesser error between actual and predicted outputs than the conventional backpropagation technique. The study thus proved that the methodology is effective in predicting uplift pressure under the diversion dams.

**Yinghua and Chang (2010)** studied the monitoring data of an arch dam to model the displacement of the dam using artificial neural networks. The data of various observation points located on and within the dam structure were collected. Principal component analysis was performed to find out the most relevant parameters out of the twelve recorded parameters. Two critical parameters were selected and by varying the hidden layer neurons, a final architecture of 2-9-1 was adopted for modeling purposes. The weights of the neural network were optimized by, applying genetic algorithms during the training phase of the neural network. The backpropagation algorithm was initialized with these optimized weights and trained further to match the actual output. The hybrid neural network model was able to converge in only 671 epochs as against 2210 epochs taken by the traditional backpropagation algorithm. The accuracy of prediction of the hybrid genetic algorithm-neural network model was compared with stepwise regression. The results showed the neural network model optimized using genetic algorithm, had good potential of modeling dam monitoring data but presented limited possibilities of modeling dam displacements based on parameters related to water, temperature and age of the dam.

In one of the studies **Ni, Zhang, and Liu (2010)** used the water quality test data for the last five years of Taihu Lake, China, for building a neural network based water quality prediction model. The neural network model is constructed to draw a

functional relationship between five different water quality parameters of a month and the same parameters for the subsequent month. The neural network model thus consisted of five input neurons and five output neurons comprising five quality parameter of every month and the same quality parameters relating to the next month. The hidden layer neurons and optimum values of weights and thresholds for the neural network were evolved using a genetic algorithm. The study showed that the genetic algorithm aided neural network provided higher degree prediction accuracy close to the actual data during training and testing of the neural network. Moreover, in comparison to conventional backpropagation learning, the hybrid genetic algorithm-neural network was able to learn the training patterns quickly. The study proved the applicability of genetic algorithm in evolving optimized neural network structure and ensuring convergence to global optimum quickly.

**Majdi and Beiki (2010)** presented the utility of stochastic search ability of Genetic Algorithms in finding the optimal number of neurons in hidden layer, learning rate and momentum coefficient of the neural network. The methodology was applied for evaluating the deformation modulus for rock masses based on the database collected from the dam sites and powerhouses located in Iran. The three important parameters for determination of deformation modulus of rock masses namely, rock mass quality designation, uniaxial compressive strength and geological strength index, were determined by performing principal component analysis on the original database. These three parameters represented the neurons for the input layer of the neural network. The output neuron of the network comprised of deformation modulus. The design of neural network architecture using genetic algorithms was started by creating a random population of chromosomes representing the neurons in hidden layer, learning rates and momentum coefficients of hidden and output neural network layers. The genetic algorithm stochastic search was able to find the optimum parameters for the neural network in just 53 generations. The prediction accuracy of the proposed hybrid neural network was compared with the neural network evolved using trial and error procedure. The results of the study demonstrated that the genetic algorithm evolved optimal neural network parameters resulted in greater prediction accuracy of the neural network model.

**Sedki *et al.* (2009)** applied real coded genetic algorithms for evolving neural network weights to cover up the limitation of the backpropagation algorithm's

entrapment at local minima. The methodology was applied to the semi-arid catchment region of Morocco. The daily rainfall-runoff data of last four years (2000-2003) was collected for this purpose. The entire data was split into two parts, namely training data, comprising of daily rainfall-runoff data of the year 2000-2002 and testing data, comprising of daily rainfall-runoff data of the year 2003. The neural network architecture consisted of for input neurons comprising of rainfall and runoff for past four days. The expected run-off for the day represented the neuron in the output layer of the neural network. The hidden layer neurons were varied, and the optimal numbers of neurons were selected by adopting trial and error. The genetic algorithm was hybridized with the neural network for determining the initial weights, and subsequently, these weights were adjusted using backpropagation algorithm to create a trained neural network. The prediction accuracy of the hybridized neural network model was compared with backpropagation neural network using the root mean square error (RMSE) and coefficient of determination ( $R^2$ ) value. The results demonstrated that the proposed methodology gave consistent and improved predictions than the traditional backpropagation neural network. The study proved the effectiveness of neural network model trained using genetic algorithms for modeling the complex rainfall-runoff phenomenon.

**Zhang and Wang (2008)** amalgamated the global search ability rendered by genetic algorithm (GA) and the local search ability rendered by the backpropagation (BP) neural network for determining the probability of earthquake occurrence based on the factors affecting the likely hood of earthquakes in a particular region. A three layered neural network was constructed with seven input neurons, 15 hidden layer neurons, and one output neuron. The training of neural network was accomplished using the backpropagation algorithm and the maximum, and minimum thresholds of the interconnecting weights were evaluated. The GA stochastic search is then employed for optimizing the weights in the search region specified by the maximum and minimum threshold values. The ANN trained using the BP algorithm was then compared with ANN optimized using GA. The trained models were applied to 5 sample data to evaluate their prediction accuracies. By employing this methodology, the learning speed of the backpropagation neural network is increased, and the best neural network architecture is evolved. The study also showed that the hybrid model

provided the higher prediction accuracy and greater consistency in the prediction of earthquakes.

In one of the studies, **Nasseri *et al.* (2008)** coupled multi-layer perceptron neural networks trained using backpropagation algorithm with genetic algorithms for forecasting rainfall. The study incorporated the utilization of genetic algorithms for deriving the optimal rain-gauge stations in the neighborhood and their lag times for prediction of rainfall. The amalgamation of genetic algorithm with neural networks led to the selection of input parameters affecting the rainfall and extracting information regarding the spatial distribution of the rainfall. Simulation studies were performed to validate the prediction accuracy of the trained neural network. For this purpose, seven models with different lag times were examined. Sensitivity analysis was conducted on the seven models using discrete and cumulative rainfall data for finding out the effect of the lag times. The methodology presented for the short term rainfall provided accurate predictions and was effective in reducing the neural network complexity. A comparison of this method with conventional neural network showed improvement in rainfall prediction accuracy. Reduction in input parameters and identification of the major lag time was also achieved through sensitivity analysis. The methodology, therefore, proved effective in modeling highly complex phenomenon of rainfall.

**Sudarsana Rao and Ramesh Babu (2007)** hybridized neural networks with genetic algorithms for developing a tool for designing of beams subjected to bending and shear. The training patterns for the neural networks were collected from the design experts. The design examples consisted of applied moments, shear and breadth of the beam in the range 30 kN-m to 125 kN-m, 30 kN to 120 kN and 250 mm to 350 mm respectively. Two different grades of concrete namely M20 and M25 were considered for the design purpose. The design output comprised of beam depth, the area of steel reinforcement and spacing of shear stirrups. The neural network was prepared by assuming five input neurons namely, moment, shear, the characteristic compressive strength of concrete, grade of steel reinforcement and breadth of beam. The output neurons of the network consisted of an area of reinforcement, the spacing of shear stirrups and depth of the beam. A trial and error technique was adopted for finalizing the optimal number of hidden layer neurons. The genetic algorithm in the study was used for optimizing the neural network weights for improving the quality of

the solution. The weights of the neural network were coded as chromosomes and were subjected to genetic operators namely, selection, crossover, and mutation for evolving their optimal values. It was found that the hybrid neural network has been able to learn the design problem in just 1000 epochs. The results showed that the hybrid neural network methodology was efficient in predicting the area of steel reinforcement, the spacing of shear stirrups and depth, close to the values given by the design experts. The study thus proved the effectiveness of the trained hybrid neural network in providing the safe design of beams subjected to bending moments and shear without performing multiple design iterations or referring to design codes or charts.

An auto-design of neural networks through genetic algorithms was proposed by **Sahoo and Maity (2007)** for assessing the structural damage. The exemplar patterns for the neural network training were simulated through finite element analysis. The exemplar patterns included frequencies and strains as inputs and the location and amount of damage to the structure as outputs. The hidden layer neurons, learning rate, and momentum factor were coded as chromosomes of genetic algorithms. The stochastic search of genetic algorithms was applied to evolve the hidden layer neurons, learning rate and momentum coefficient. The hybrid genetic algorithm-neural network methodology was implemented for two structures namely, clamped-free beam and plane frame. The study showed that the method provided good prediction accuracy with percentage error in the range of 2.5 % to 2.8%.

**Srinivasulu and Jain (2006)** used three different training methods for neural network modeling of rainfall-runoff. They investigated and compared the effectiveness of backpropagation, real coded genetic algorithm, and self-organizing map neural network training methodologies. Three different models based on these methodologies were created by using stream flow data and the daily rainfall derived from Kentucky River basin. In the first model, the architecture of backpropagation neural network was determined using trial and error by altering the number of hidden layer neurons. The second model was created by using the real coded genetic algorithm to train this neural network architecture. Finally, for the third model self-organizing map was used for drawing the rainfall-runoff relationships. The performance of these models was compared using seven different statistical performance metrics. The comparative study showed that neural networks trained using real coded genetic algorithm were able to outperform the prediction accuracy of

backpropagation neural networks and self-organizing maps. Also, the neural networks trained using real coded genetic algorithm was able to provide consistent predictions during training and testing phases.

In one of the studies undertaken by **Sudrasana Rao *et al.* (2006)**, the stress-strain response of ceramic-matrix-composites comprising of aluminum oxide and silicon carbide were simulated using a neural network. The results of the finite element model for the ceramic-matrix-composite as suggested in earlier studies were used for development of neural network training patterns. A neural network model comprising of four input neurons namely, interface strength, failure mode, strain level and square of interface strength along with output neuron corresponding to stress level was prepared. The hidden layers and corresponding hidden layer neurons of the neural network were arrived by trial and error process. A two hidden layer neural network was selected with sigmoidal transfer functions. Genetic algorithm was utilized for evolving the initial weights for backpropagation algorithm training of the neural network. This hybridization was done to minimize the training cycles required by the neural network. The results showed that the hybrid neural network has been able to learn quickly in just 2000 cycles as compared to 32000 cycles required by the conventional backpropagation algorithm. The trained neural network effectively learned the nonlinear stress-strain relationships and was subsequently tested for the patterns not included earlier during the training phase. The hybrid neural network was able to predict the stress-strain behavior close to the finite element analysis and can, therefore, the methodology can considerably reduce the computational effort in arriving at the stress-strain behavior. The study showed that the genetic algorithm based backpropagation model has the ability to minimize the requirement of handling large volume of finite element analysis data and therefore can be effectively implemented on a personal computer without requiring a larger memory.

**Wu and Chau (2006)** hybridized genetic algorithm with the artificial neural network model for forecasting flood in Yangtze River of China. The model was used for forecasting of water levels at the downstream of a station on the basis of water levels recorded at the upstream of the recording station. A total of 1456 input-output datasets was collected and divided into three datasets, namely training, validation and testing datasets. The genetic algorithm was applied for computing the neural network optimal weights and threshold values. The hybrid neural network model was

compared with the neural network, genetic algorithm and linear regression models. The study demonstrated that the hybrid neural network model was able to deal with the over-fitting problem of neural networks, and better prediction accuracy was achieved at a faster convergence. However, this was achieved at the expense of higher computational time taken by the hybrid neural network. However, the study proved the effectiveness of hybrid neural network in accurate prediction of the flood of the channel reach between the two stations.

**Sudarsana Rao and Ramesh Babu (2006)** hybridized neural networks with genetic algorithms for modeling percentage reinforcement of short columns subjected to biaxial bending. The backpropagation neural network was used to model the percentage reinforcement of the short column based on eight design inputs namely, axial load, biaxial moments (moments in two directions), breadth and depth of column, concrete grade, grade of steel reinforcement and effective cover in both directions. The design experts were presented with different problems about biaxial column design. Adopting this methodology, 220 exemplar patterns for neural network training were generated. The neural network architecture with five hidden layer neurons was selected by trial and error process. Genetic algorithms were harnessed for evolving the neural network synaptic weights to mitigate the possibility of backpropagation neural network entrapment at a local minimum. The study demonstrated that by the hybridizing neural network with a genetic algorithm, the speed of learning can be enhanced. Moreover, the percentage reinforcement predicted by the trained hybrid model was found to be in close agreement with that obtained by the interior penalty function optimizer and, therefore, proved that neural networks can be used for safe and economical of columns subjected to biaxial bending.

In one of the studies by **Kim et al. (2004)**, the historical data about construction costs of 530 residential building projects were used for creating an artificial neural network based cost estimation model. A backpropagation neural network utilized for this purpose comprised of twelve input neurons and one output neuron. The neurons in the input layer consisting of the parameters affecting the total cost of the projects and output neuron represented the actual cost of the project. The genetic algorithm was used for determining the number of hidden layer neurons, momentum coefficient, and learning rate. A two hidden layer artificial neural network was able to provide better prediction than one or three hidden layers neural network.



The neural network comprising an optimal number of hidden layer neurons and training parameters evolved through genetic algorithms, was able to provide greater predictive accuracy than neural network evolved through trial and error. The study proved the effectiveness of the hybridized genetic algorithm-neural network in the development of a cost estimation model at the early stage of the project execution.

The strength of lateritic soils was modeled by **Attoh-Okine (2004)** using a hybrid genetic algorithm-neural network methodology. For the development of the strength model, tests were conducted on the lateritic soils located in Ghana. The test values corresponding to field dry density, relative compactness, field moisture content, lab maximum dry density, lab optimum moisture content, plasticity index, liquid limit and field California bearing ratio (CBR) were collected from 45 pavement test location sites. The genetic algorithm was amalgamated into the training phase of a neural network for evolving optimized neural network weights. Five different neural network architectures were developed with a various set of input neurons. The CBR value formed the output neuron for the neural network. It was found that field density was the single most important parameter influencing the CBR value of the lateritic soil. The study showed that for lateritic soils, the CBR value bears a simple correlation to the field density and hence this information can be harnessed for easy determination of CBR value of pavements resting on lateritic soils. The study also showed that the neuro-genetic model was more efficient in identifying the critical inputs for determination of CBR value.

A methodology for prediction of crack width in precast jointed reinforced concrete beams was developed by **Avila *et al.* (2004)**. Two distinct modeling approaches were used, one using backpropagation and the other incorporating the stochastic search of genetic algorithms for the training of neural networks. A series of two beams with different geometrical properties, reinforcement, and anchor bar configuration were loaded vertically. The load was increased in increments till failure of the beam was noticed. The crack width was evaluated at two upper limits of tensile stresses namely, 200 MPa and 300 MPa respectively. A total of 24 beams were tested, and the experimental results were considered for generating the data for the neural network modeling. The values of two input neurons were evaluated using empirical relationships based on anchor bar configuration and spacing of reinforcement bars. The maximum crack width formed the output neuron for the neural network model.

The neural network training was performed using two different training algorithms namely, backpropagation and genetic algorithms respectively. The results showed that, by optimizing connection weights through genetic algorithms, a significant reduction in prediction error can be achieved. Regardless of the fact that the genetic algorithm training required more time in comparison to backpropagation algorithm training, the accuracy derived from the genetic algorithm training of neural network outperformed the performance of the backpropagation algorithm.

### 3.3.3 Multidisciplinary applications

Many applications have hybridized GA with conventional backpropagation neural networks (BPNN). Some of the notable recent applications harnessing the methodology include, detection of status of rotary components in agro-industrial machines (**Martinez-Martinez *et al.*, 2015**), evaluating bridge health using results from fiber Bragg grating sensor (**Ai and Guo, 2014**), predicting water stage during typhoon events (**Liu and Chung, 2014**), classification of tea specimens (**Plawiak and Maziarz, 2014**), prediction of fracture in low-permeability reservoirs (**Xue *et al.*, 2014**), classification model for cotton yarn quality (**Amin, 2013**), correlation of density in nanofluids (**Karimi and Yousefi, 2012**), forecasting ozone concentration (**Feng *et al.*, 2011**), and prediction of saturates of sour vacuum gas oil (**Wang *et al.*, 2010**).

During last few decades, a number of improved training algorithms namely, Levenberg-Marquardt (LM) backpropagation algorithm, Resilient backpropagation, Scaled Conjugate Gradient, One Step Secant, etc. Amongst these, the LM backpropagation algorithm owing to its faster convergence has been widely used with GA for training the Multilayer Feedforward Neural Network (MFNN). Some of the notable recent multidisciplinary implementations of hybridizing GA with the Levenberg-Marquardt backpropagation neural networks (LMBNN) include, approximation of phenol concentration (**Plawiak and Tadeusiewicz, 2014**), hypoglycemia detection using EEG signals (**Nguyen *et al.*, 2013**), automotive price forecasting (**Peyghami and Khanduzi, 2012**), stock exchange index prediction (**Asadi *et al.*, 2012**), and determination of dimensions of asymmetric coplanar waveguide (**Wang *et al.*, 2012**).

### 3.4 Conclusions of the literature survey

The broad conclusions derived from the literature survey are summarized under:

- a) The backpropagation (BP) algorithm trained Multilayer Feedforward Neural Networks (MFNN) are good function approximators. They have been extensively harnessed for tasks associated with modeling of physical phenomenon and material behavior, wherein conventional regression models do not yield the desired accuracy and predictability.
- b) All previous studies related to modeling slump of concrete have harnessed BP algorithm trained MFNN.
- c) In most of the past studies, the optimal number of hidden layers and their neurons for neural network model have been determined by adopting trial and error technique.
- d) The neural network training is accomplished by iterative updating of neural network weights and biases through backpropagation of error to reduce the error between the actual and predicted outputs to an acceptable minimum. The error surface of any function required to be approximated contains many local minima and one global minimum. The steepest gradient descent methodology of backpropagation algorithm requires initial weights and biases to be located near global minimum, to allow its faster convergence. Conversely, initial weights and biases located near a local minimum increase the probability of the algorithm getting trapped in a local minimum and eventually would either never reach the global minimum or would require a significant number of iteration cycles to reach the desired performance goal. The accuracy and the convergence rate of the neural networks trained using backpropagation algorithm thus depend to a large extent on the initial weights and biases.
- e) Hybridization of GA with the MFNN trained using backpropagation algorithm improved the prediction accuracy of MFNN and enabled faster convergence to the global optimum. This improvement is largely attributed to the initial neural network weights and biases optimized using the global search rendered by GA. The backpropagation algorithm initialized using

optimal weights and biases were able to improve the accuracy the MFNN and quickly converged to the global minima. Thus, hybridization of GA with BPNN was able to cover up the inherent drawback of the backpropagation algorithm.

- f) A survey of literature has revealed that very few applications related to modeling properties of concrete have utilized the hybrid Genetic Algorithms-Artificial Neural Networks (GA-ANN) methodology.
- g) Among the various improved backpropagation training algorithms like Resilient Backpropagation, BFGS Quasi-Newton, One Step Secant and Scaled Conjugate Gradient, the Levenberg-Marquardt (LM) backpropagation algorithm provides both prediction efficiency and fast learning to the MFNN. In most of the applications, the LM algorithm has been hybridized with GA to improve the convergence speed and prediction accuracy of the trained neural network.

### **3.5 Research gap**

The literature survey has revealed that MFNN trained using backpropagation algorithm has been the preferred choice for the researchers for modeling unknown, complex or nonlinear functional relationships related to either physical phenomenon or material behavior. However, it has certain drawbacks, which can be covered up by adopting a hybrid methodology to train the MFNN. A survey of the literature has shown that extensive civil engineering and multidisciplinary applications have harnessed the hybrid GA-ANN methodology. Despite numerous applications of hybrid GA-ANN, very few studies in the past have employed GA-ANN methodology for modeling material behavior of concrete. Moreover, this methodology has not been so far employed for modeling slump of concrete. Apart from the modeling methodology, the relative importance of concrete's design mix constituents on the concrete slump value using the weights of the trained neural network model has not been quantified so far.

## **Chapter 4**

### **Modeling and Analyzing Concrete Slump**

---

- **Introduction**
  - **Material**
  - **Methodology of modeling slump of RMC using hybrid Genetic Algorithms-Artificial Neural Networks**
  - **Modeling slump of RMC using first order and second order regressions**
  - **Evaluating performance of the models**
  - **Sensitivity analysis**
  - **Assessing applicability of the developed slump model for a different RMC batching plant**
  - **Developing decision support tool for estimating concrete slump**
  - **Summary**
-

## 4.1 Introduction

Workability is an important physical property of fresh concrete that governs the ease with which concrete can be placed and compacted at the site with sufficient resistance to bleeding and segregation. This property is primarily governed by three factors namely, the rheological properties of the cement paste which act as a lubricant, the intermolecular friction of the aggregate particles and the friction between the fresh concrete and the formwork surface during its placement at the site (Yeh, 2007). Workability represents the internal work required to overcome these frictional forces to achieve full compaction of concrete. There is no direct test for measuring the workability in terms of the mechanical work. However, an indirect test known as slump test is commonly used in the laboratory and site for assessing the consistency and homogeneity of concrete. In actual practice, the slump test is repeated with different trial mixes proportions for designing a concrete catering to specific workability requirement. This experimental procedure is time-consuming. Moreover, traditionally used empirical equations do not provide an accurate assessment of the interactions between the different constituents and their overall effect on the slump of concrete. Thus, the complexity of material behavior exhibited by concrete due to a large number of constituents exhibiting different property necessitates a non-algorithmic approach to material modeling.

Artificial Neural Networks (ANN) is a biologically inspired computing tool that has been widely used by researchers for modeling unstructured problems. Based on the information processing paradigm, ANN derives immense learning ability by simulating the connectionist architecture of the human brain. The basic strategy for developing ANN based material models is to train the network with an adequate number of experimental results. The adaptive nature of these models re-organizes its architecture to imbibe the complex relationship underlying the interactions between the various material constituents and the property defining the material behavior. It is brought about by a training algorithm that updates the weights and biases through the forward propagation of information and backpropagation of errors.

Among the various architectures of ANN, the Multilayer Feedforward Neural Network (MFNN) trained using backpropagation (BP) algorithm has proved to be the most versatile tool for tasks associated with function approximation and modeling of material behavior. However, the accuracy and convergence speed of MFNN model is

heavily dependent on the initial value of weights and biases for BP algorithm. To cover up the inherent drawback of BP algorithm, the stochastic search of genetic algorithm (GA) through simulation of the biological evolutionary processes is utilized for determining the optimal initial value of weights and biases for the BP algorithm to, assist effective modeling of concrete's material behavior.

The study deals with the modeling slump of Ready Mix Concrete (RMC) based on the proportions of its design mix constituents using hybrid GA-ANN methodology. The sub-sections of the chapter comprehensively deal with the collection of exemplar data for modeling concrete slump, preparing training, validation and test datasets, normalization of data, developing the neural network architecture, hybrid methodology for modeling the slump of ready mix concrete, developing regression-based models for concrete slump, comparing the performance of the slump models, quantifying the importance of various constituents of RMC on the slump value, assessing the universal applicability and accuracy of the trained hybrid neural network model for predicting slump for the design mix proportions collected from a different RMC plant and developing a decision support tool to aid quick estimation of concrete slump.

## **4.2 Material**

### **4.2.1 Collection of data and its description**

The exemplar data for developing a neural network model for concrete slump were collected from a particular RMC batching plant to minimize the error caused due to variations in the physical and chemical composition of the raw materials used in manufacturing the ready mix concrete. The data constituted the design mix proportions of concrete and their slump value. In all a total of 493 design mix proportions of concrete grade M10 to M35 comprising of concrete constituents namely, cement, pulverized fuel ash, sand, coarse aggregate (20mm), coarse aggregate (10mm), superplasticizer and water in  $\text{kg/m}^3$ , along with the slump value in mm were collected from the first batching plant. The data was used for the developing the model for the concrete slump. The details and statistics of the data utilized for model development are exhibited in **Table 4.1**.

**Table 4.1:** Details and statistics of the exemplar RMC data used for modeling slump

RMC data	Statistics				Specific Gravity
	Min.	Max.	Mean	SD	
Cement (kg/m <sup>3</sup> )	100	450	273.0406	73.5491	3.15
Pulverized fuel ash (kg/m <sup>3</sup> )	0	200	62.1440	52.1081	2.10
Sand (kg/m <sup>3</sup> )	472	900	763.7363	53.4742	2.66
Coarse aggregate 20mm (kg/m <sup>3</sup> )	412	764	576.9736	52.1550	2.65
Coarse aggregate 10mm (kg/m <sup>3</sup> )	343	682	532.7728	46.5083	2.65
Superplasticizer (kg/m <sup>3</sup> )	0.00	5.80	3.7072	0.6767	1.15
Water (kg/m <sup>3</sup> )	105	190	166.6227	8.7324	1.00
Slump (mm)	90	190	157.7992	8.2315	-

### 4.3 Methodology for modeling slump of RMC using hybrid Genetic Algorithms-Artificial Neural Networks

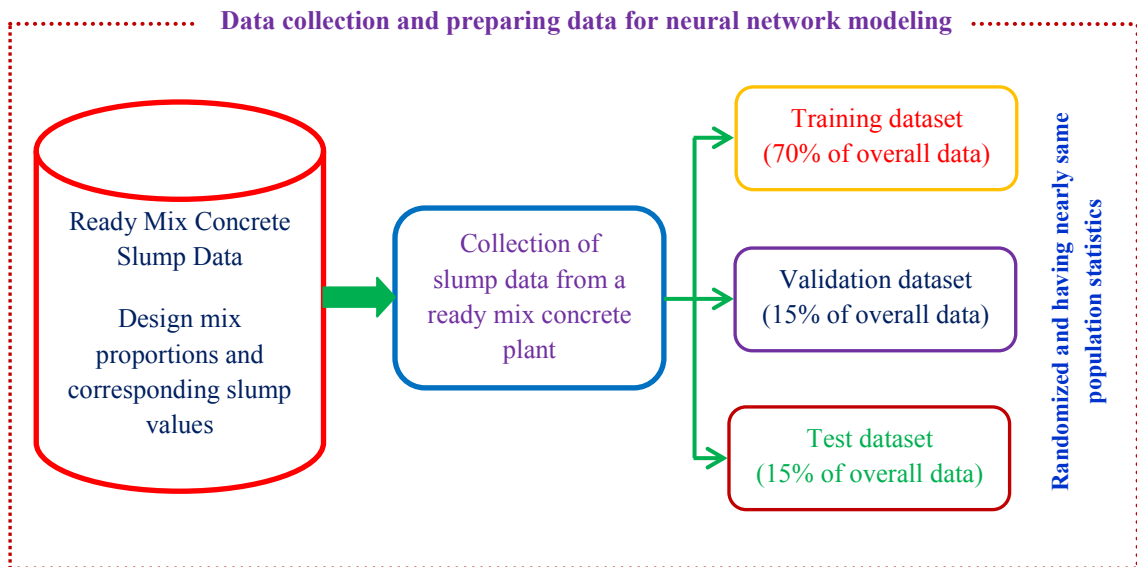
#### 4.3.1 Splitting data into training, validation, and test datasets

The neural networks imbibe the information processing characteristics of the human brain through a network of closely connected computational units called the artificial neurons. In supervised learning, the neural networks are trained to learn from examples and are subsequently validated to assess their generalization ability. Once trained, the neural networks are tested for evaluating the network error. Therefore, splitting of the available data into three parts is a major step in the development of neural network model (Maier *et al.*, 2010). However, there are no specific rules regarding the percentage of data contained in these subsets. In the present study, 70% of the available or overall data are allocated for training and the remaining 30% are equally divided for performing validation and testing of the neural networks.

The training data set is presented in the form of matched input-output configuration, to allow the systematic updating of weights and biases of the neural network model. Once the network error, measured in terms of squared error between the actual and predicted outputs reaches a threshold limit, the neural network is considered as trained. This trained neural network may or may not yield the same



prediction performance or generalization when presented with an unseen data set. This phenomenon is attributed to over learning or overfitting of the training data set by the neural network. To strike a balance between the learning and the generalization of the neural network, validation error is monitored at each iteration cycle by indirectly incorporating the validation data set during its training. The early stopping of the neural network training is undertaken, if the validation error begins to rise. The information in the form of data available in the training, validation, and test dataset, therefore, influence the learning ability of the neural network during the training phase, its generalization when presented with unseen data and the overall performance of the trained neural network model respectively. The importance of data division is, therefore, necessitated for the effective mathematical modeling of a phenomenon using ANN. The data collection and division of data for the present study is schematically shown in **Figure 4.1**.



**Figure 4.1:** Data collection and preparing data for neural network modeling

The amount and the nature of data or information comprising the training data set are critical for the successful performance of the neural network, due to its inability to extrapolate the predictability beyond the data range used during training (Flood and Kartam, 1994; Tokar and Johnson, 1999). Masters (1993) also concluded that if the training data set is not completely representative of the population, it may lead to neural network overfitting. Hence, the training data set should be large enough (Flood and Kartam, 1994; Ray and Klindworth, 2000) and must extend to the edges of the modeling domain to incorporate all possible variations

in the available data (**Bowden et al., 2002; Kamp and Savenije, 2006**). Therefore, it is ensured that the minimum and maximum values for each exemplar pattern are included in the training dataset, and the entire data is divided into three statistically similar and representative subsets such that, their statistics namely, mean and standard deviation (SD) are marginally different from that of the population (**Table 4.2**).

#### 4.3.2 Data normalization

The activation function or transfer function introduces non-linearity in the multi-layer perceptrons (**Arbib, 1995**) and augments the ability of the network to robustly deal with input-output relations that are either undefined or complex in nature (**Shamseldin et al., 2002**). The non-linear activation functions commonly used for the backpropagation neural network are logistic sigmoid or tangent hyperbolic. The output range of these activation functions is bounded and, therefore, demands pre-processing of the training data so that they fall within the minimum-maximum range of the activation function. An individual scaling of input and output patterns is usually done to maximize the variance in the available data (**Kalogirou, 2003**) for obtaining good results and significantly reduces the computation time (**Nawi et al., 2013; Sola and Sevilla, 1997**). The data normalization also helps the neural networks to efficiently learn features comprising of different identities by minimizing the bias within the network towards a particular feature (**Priddy and Keller, 2005**), to ensure that all features get same significance during the training phase (**Maier and Dandy, 2000**).

In the present study min-max normalization given by **Equation 4.1** having the advantage of preserving the distribution of the corresponding features has been used to linearly transform the data to fall within a defined range [-1, 1].

$$p_{norm} = \frac{2 * (p - p_{min})}{(p_{max} - p_{min})} - 1 \quad (4.1)$$

where,  $p_{norm}$  is the normalized value of the variable  $p$  whose maximum value is  $p_{max}$  and the minimum value is  $p_{min}$ . After neural network is trained, validated and tested, the values are de-normalized to the actual values using:

$$p = \frac{1}{2} (p_{norm} + 1) (p_{max} - p_{min}) + p_{min} \quad (4.2)$$

**Table 4.2:** Statistical parameters of training, validation, and test datasets

RMC data constituents	Training (346 datasets)				Validation (74 datasets)				Test (73 datasets)			
	Min.	Max.	Mean	SD	Min.	Max.	Mean	SD	Min.	Max.	Mean	SD
Cement (kg/m <sup>3</sup> )	100	450	272.2138	73.2201	120	450	276.1351	76.6385	100	450	273.8219	72.9490
Fly Ash (PFA) (kg/m <sup>3</sup> )	0	200	62.9046	52.3338	0	160	60.3243	52.0189	0	120	60.3835	51.7476
Sand (kg/m <sup>3</sup> )	472	900	764.7062	53.6481	546	870	760.8919	54.8234	546	860	762.0274	51.8242
Coarse Aggregate 20mm (kg/m <sup>3</sup> )	412	764	576.9595	52.8755	438	764	578.5000	49.9636	486	730	575.4931	51.5305
Coarse Aggregate 10mm (kg/m <sup>3</sup> )	343	682	532.7312	47.6015	428	680	533.3243	45.8269	408	593	532.4109	42.3496
Superplasticizer (kg/m <sup>3</sup> )	0.00	5.80	3.6934	0.6659	1.30	5.50	3.7731	0.6365	1.50	5.50	3.7059	0.7668
Water (kg/m <sup>3</sup> )	105	190	166.3815	8.8421	150	186	168.6216	8.2340	150	185	165.7397	8.5163
Slump (mm)	90	190	157.6734	8.2272	115	190	158.6486	8.1214	115	170	157.5342	8.4213

### 4.3.3 Neural network architecture and training parameters

Artificial neural networks are a simplified abstract of a human brain, as they learn from examples and can store the acquired knowledge through inter-neuron synaptic connection value denoted as weight. The artificial neuron carries an additional input called bias  $b$  which is interpreted as an additional weight. The bias has a constant value and plays a major role in the efficient learning of the neural network. The artificial neurons form the computational processing units for neural network model. The architecture of an artificial neural network (ANN) is defined by the interconnection of the neurons arranged in layers, a learning algorithm for systematic updating and adjusting of weights and biases and an activation function. MFNN is the most commonly used neural network architectures. The artificial neurons arranged in layers designated as “input layer,” “output layer” and a number of intermediate “hidden layers”, form the topology for a MFNN. This type of topology permits only inter-layer connections among the artificial neurons in the forward direction only. The processing units called the neurons receive the input signals from their neighboring neurons. The signal is processed through a transfer function, and the output is propagated to the next layer.

The input layer neurons help in feeding the information into the neural network. The information is propagated from the input layer neurons to the output layer neurons through a series of hidden layer neurons through the inter-layer neuron links called the synaptic weights. A positively valued weight propagates the signal in the forward direction, whereas a negative value inhibits the signal. The output produced by the neural network received at the output layer neurons is compared with the actual or target value. The weights and biases are repetitively adjusted to render the computed network error to an acceptable minimum. The updating and adjusting of the weights and biases of the neural network are brought about by a learning algorithm. The error backpropagation or simply backpropagation (BP) algorithm is the most widely used training algorithms due to its simple implementation. The BP algorithm comprises of forward propagation of information for computing network error for a given set of weights and biases and a back propagation of this error for modifying weights and biases. The BP algorithm employs the principle of steepest gradient descent to adjust the neural network weights and biases iteratively to reduce the network error to an acceptable minimum.

The MFNN trained using the BP algorithm, are a popular choice for the researchers in dealing with tasks associated with the mapping of nonlinear functional relationships. In the study, the MFNN has been created using the Neural Network Toolbox included in the commercially available software MATLAB 7.1 (R14 SP3) (Version 7.1.0.246). The design mix proportions of the RMC namely, cement, pulverized fuel ash (PFA) or fly ash, sand, coarse aggregate (CA) 20mm, coarse aggregate (CA) 10mm, superplasticizer (SP) and water, form the seven input neurons of the input layer. The slump value of concrete is designated as the output layer neuron. The hidden layers and hidden layer neurons are needed in the neural network for improving the learning and generalization ability of the neural network and are thus governed by the complexity of the function to be approximated, the number of training cases and the inherent noise in the data (Sovil *et al.*, 1997). Too many hidden layers and/or hidden layer neurons over-trains the neural network, enhancing its memorizing power and significantly affecting its recognizing ability, leading to poor generalization.

Studies conducted in the past for determining the optimal number of hidden layers have indicated that, any complex non-linear function can be approximated to an acceptable degree of accuracy by a single hidden layer of ANN (Cybenko, 1989; Funahashi, 1989; Hornik *et al.*, 1989; Jalili-Gazi Zade and Noori, 2008; Noori *et al.*, 2009; Noori *et al.*, 2010) and an increase in the number of neural network hidden layers may not result in significant performance improvements (Patuwo *et al.*, 1993). Numerous studies for determining the optimal number of hidden layer neurons for neural network architectures have been performed (Hunter *et al.*, 2012; Jinchuan and Xinzhe, 2008; Molga, 2003; Pendharkar and Rodger, 2003; Shibata and Ikeda, 2009; Tamura and Tateishi, 1997; Zhang *et al.*, 1998 ). Despite numerous “rules of thumb”, the number of hidden layer neurons are decided by a trial and error procedure, starting with a minimum number and gradually increasing till no further improvement in the generalization of the network is noticed (Nawari *et al.*, 1999; Poulton, 2002).

For a trained neural network model, ideally the network error i.e., difference between target or actual value and the model predicted value,  $e_i = (T_i - P_i)$  for all observations should be preferably close to zero. In such a case, the average network error measured in absolute term shall be close to zero. Moreover, the standard

deviation of the network error shall be small and errors shall be confined in close vicinity of the mean error.

In view of the above, eleven different topologies of the neural network having one hidden layer and neurons in the range 4 to 15 were examined ten times for evaluating the optimal number of hidden layer neurons and their learning and generalization ability have been assessed using two performance metrics namely, standard deviation (SD) of errors and mean absolute error (MAE). The standard deviation (SD) of error for each trial is computed using **Equation 4.3**. The average standard deviation of errors (AVSD) is evaluated for ten successive trials using **Equation 4.4**. The mean of absolute error for each trial is computed using **Equation 4.5**, which is subsequently averaged for ten successive trials using **Equation 4.6** for evaluating the average mean absolute error (AVMAE). This methodology was implemented on both training and validation datasets. The topology of the neural network, which gave the least value of the performance metric during the validation phase, was selected for the modeling purpose.

$$SD(e) = \sqrt{\frac{\sum_{i=1}^N (e_i - \bar{e})^2}{N-1}} \quad (4.3)$$

$$AVSD = \frac{\sum SD(e)}{N_{Trials}} \quad (4.4)$$

$$MAE = \frac{\sum_{i=1}^N |e_i|}{N} \quad (4.5)$$

$$AVMAE = \frac{\sum MAE}{N_{Trials}} \quad (4.6)$$

where,  $T_i$  is the target or actual value and  $P_i$  is the predicted value of the slump for the  $i$ th data,  $e$  is the network error  $\bar{e}$  is the mean of the network error,  $N$  is the number of data patterns and  $N_{Trials}$  is the number of trials. In the present case  $N_{Trials}$  is taken as 10.

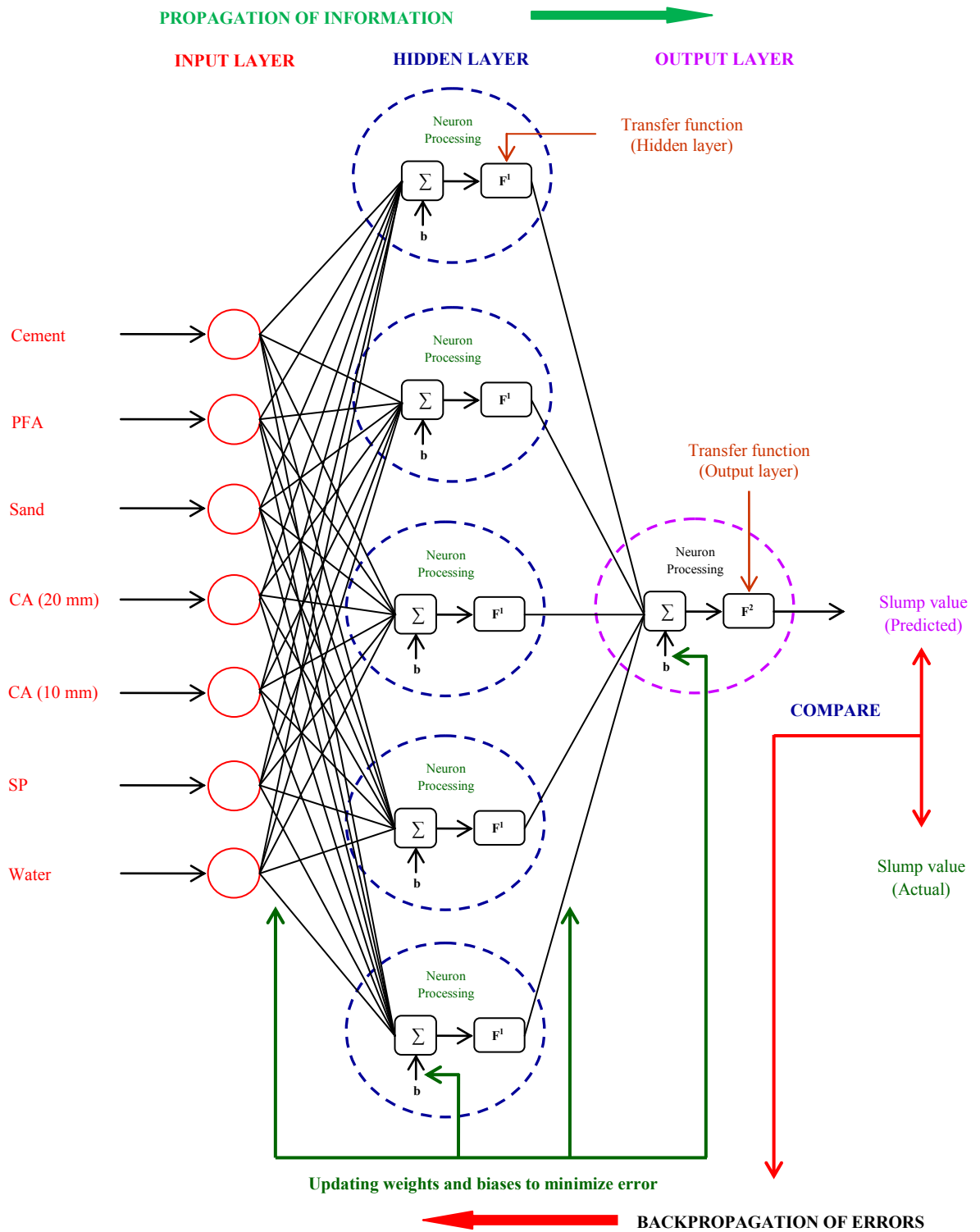
The initial range of neural network weights and biases has a significant effect on the convergence speed. **Sietsma and Dow (1991)**, **Gallagher and Downs (1997)**, and **Staufer and Fisher (1997)** suggested that the initial weights and biases should be

initialized to small random values. In the present study, before training the neural networks, the weights and biases were randomized and initialized in the range  $[-0.5, 0.5]$ . This is primarily done to break the symmetry (Li *et al.*, 1993) and to keep the network out of unstable equilibrium caused due to initialization of neural network with the same values of weights and biases. Gomes *et al.* (2011) concluded that the performance and the computing time of the neural network are also influenced by the type of training algorithm employed for iterative updating of weights and biases and the kind of transfer function.

A number of modified versions of the standard backpropagation algorithms are available for training the MFNN. In contrast to the other modifications of backpropagation algorithms, the Levenberg-Marquardt (LM) algorithm harnesses a second order nonlinear optimization technique (Adamowski *et al.*, 2012) that combines the speed of Gauss-Newton algorithm and the stability rendered by the steepest descent algorithm (Wilamowski *et al.*, 1999). The advantages of faster convergence and efficiency to seek good quality local minima rendered by the LM algorithm, has made this algorithm a first choice among the researchers for supervised training of neural networks. The present study uses the Levenberg-Marquardt (LM) backpropagation algorithm for the systematic updating of the neural network weights and biases for MFNN. This neural network is hereafter called the Levenberg-Marquardt backpropagation neural network (LMBNN).

In most of the tasks associated with ANN modeling, a certain degree of non-linearity is essential, which is rendered by introducing a differentiable transfer function (Zhang, 1998). Among the numerous non-linear transfer functions, the logistic and tangent hyperbolic transfer functions are generally used for applications associated with function approximation (Zheng, 1999). The logistic and tangent hyperbolic transfer functions squashes the inputs in the range  $[0, 1]$  and  $[-1, 1]$  respectively. The tangent hyperbolic function because of a broader range in comparison to logistic function is shown to provide a greater response to a small deviation in the input. Moreover, the tangent hyperbolic transfer function is shown to render faster convergence of learning algorithms (Bishop, 1995; Karlik and Olgac, 2011; Ozkan and Erbek, 2003). Owing to the advantages exhibited by the tangent hyperbolic function, this transfer function has been used in the hidden layer, and a linear transfer function has been used for the output layer to

facilitate comparison of actual and predicted slump values. The flowchart depicting the typical neural network architecture for the study with five hidden layer neurons trained using backpropagation algorithm is shown in **Figure 4.2**.



**Figure 4.2:** Training of MFNN using backpropagation algorithm



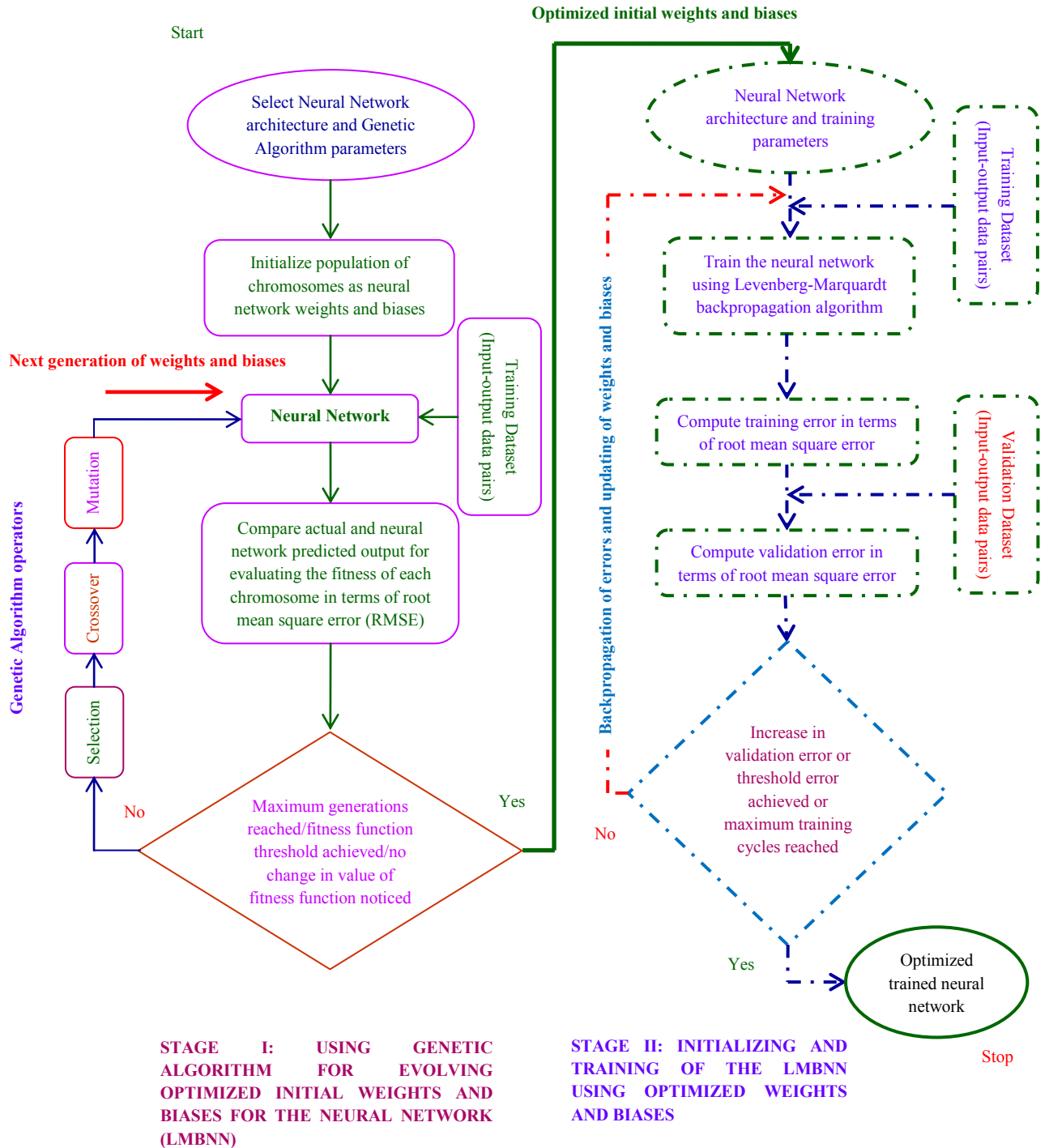
#### 4.3.4 Neural network optimization using genetic algorithms

Genetic algorithms (GA) are evolutionary search and optimization algorithms that replicate the nature's biological evolution process in a computing environment through the use of evolution and genetic operators. Inspired by "survival of the fittest" heuristic, the algorithms harnesses probabilistic operators, that work simultaneously on a number of solutions and can be applied to either mixed, discrete or continuous nature of optimization problems and therefore have a distinctly different approach in comparison to conventional optimization algorithms that employ deterministic operators, work on a single solution point at a time and cannot be conveniently modified to deal with the different nature of optimization problems (**Lagaros *et al.*, 2002**). Moreover, GA's global search amalgamates exploration and exploitation of the solution space to find a solution to complex, non-linear problems and offers gradient-free optimization, to render practical alternatives to unstructured problems wherein the gradient information is either unreliable or difficult to compute.

Although, the LM backpropagation algorithm offers fast convergence speed due to Gauss-Newton algorithm, yet its steepest gradient descent part relies on the gradient information of the error surface which is not always deterministic and therefore, its probability of reaching a global optimum and finding satisfactory solutions is not guaranteed as it is likely to get trapped in local minima in the proximity of the algorithm's starting point in the solution search space. GA on the other hand, by offering a gradient-free approach carries lesser probability of getting trapped in local minima. However, since the GA performs randomized global search, its convergence to the global optimum is quite slow. It demands a complementing hybrid methodology that can extract the benefits of the individual technique while covering up their limitations.

The hybrid methodology for modeling slump of RMC presented in the study thus comprises of two stages. The first stage utilizes the global optimization and search of GA for evolving the optimized initial weights and biases for MFNN. The Genetic Algorithm Toolbox included in the commercially available software MATLAB 7.1 (R14 SP3) (Version 7.1.0.246) has been used for this purpose. In the second stage, the LM algorithm starts with the GA evolved optimized set of weights and biases and iteratively updates and fine tunes these weights and biases to reach a

global optimum. The hybrid GA-LMBNN methodology flow chart is exhibited in **Figure 4.3**.



**Figure 4.3:** Evolving initial weights and biases for MFNN using GA and subsequent training and validation of LMBNN

Broadly, the hybrid GA-LMBNN modeling methodology depicted in **Figure 4.3** comprised of the following steps:

**1. Evolving weights and biases for ANN using GA**

**a. Initialization of GA with population of chromosomes**

The GA's stochastic search and optimization process work on a set of chromosomes that describe the possible solutions to a problem. The size of the initial population of chromosomes plays a critical role in successful convergence of GA to a global optimum as, a small sized population due to lack of enough genetic material may fail to cover the entire solution search space, and a larger population may converge to a global optimum but at the expense of greater computational time (**Khan et al., 2008**). In the light of the issues narrated above, an initial population of chromosomes is chosen to balance the performance and computational time of GA.

**b. Evaluating chromosome fitness**

The fitness of a chromosome is a measure that indicates the quality of the solution and its closeness to the optimum value (**Ben-Romdhane et al., 2013**) and thus helps in differentiating the optimal solution from numerous sub-optimal solutions. In the present study, the possible solutions to the problem namely, the weights and biases coded as the chromosomes are assigned to the neural network. The training data set comprising of  $N$  number of input-output pairs are presented to the neural network and the objective is to minimize the mean of squared errors (RMSE) computed for the actual or target slump value  $T_i$  and the ANN predicted slump value  $P_i$  given by **Equation 4.7**. The root of the mean of squared errors has been considered as the fitness function ( $F_{fitness}$ ) is given by the equation:

$$F_{fitness} = RMSE \tag{4.7}$$

$$\text{where, } RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_i - P_i)^2}$$

**c. Creating the next generation of population**

The GA models the biological adaptation process that selects the fitter chromosomes having a greater chance to reproduce, adapt and survive to the change in the environments and conditions. It is achieved using three basic stochastic operators namely, selection, crossover and mutation, synonymous to the biological operators of natural selection and genetic inheritance.

**i. Selecting the fitter chromosomes**

This operation is algorithmically achieved by using a selection operator that picks the chromosomes based on their fitness function value. The higher the fitness function value, the greater is the probability of a chromosome to be represented in the next generation of the population, thereby enriching the next generation with fitter chromosomes. Using this approach, the selection operator reduces the search area of GA within the population by, filtering out the poor solutions.

In the present study, this is achieved by using a tournament selection strategy which provides a chance to all individual chromosomes to be selected thus, maintaining the diversity in the population. A mating pool is created, and all chromosomes are inserted into this pool to compete with each other to select the winner. An increase in the size of the tournament diminishes the chance of weaker individuals to be selected. Therefore, tournament selection strategy is suitable for small sized problems (**Razali and Geraghty, 2011**).

**ii. Applying variation operators**

The variation operators used in GA can be categorized as recombination or crossover operators and mutation operators that help in creating the new population of chromosomes from old ones. These genetic or variation operators, namely crossover and mutation form the backbone of the GA and yield power to its search operations (**Lin et al., 2003**). Using these operators, the GA can concentrate on the search regions where there are greater possibilities of finding a global optimum.

The crossover operator exchanges the information of the parent chromosomes and based on the historical information that, a currently good solution was found in the region, it exploits the current population to focus on a localized region of the search space. In the present study, a scattered crossover has been adopted. If all the pairs of the parent chromosomes are used for creating offsprings, the crossover fraction ( $P_c$ ) is 1, or the probability of crossover is 100%. Past studies have suggested that the probability of crossover should be kept high (**Lin et al., 2003; Mellit et al., 2010; Senouci and Al-Ansari, 2009; Tan et al., 2014**), so that the parent chromosomes are replicated to a lesser extent in the next generation.

In contrast to the crossover operator that performs the exploitation of the current solution, the mutation operator augments the exploratory power of GA. The mutation operator flips the genetic material within the chromosome thereby, maintaining genetic diversity in the population, mitigating premature convergence and prohibiting the search process to fall into local minimum. The present study uses the uniform mutation that, replaces the value of a chosen gene with a random number uniformly distributed between upper and lower bound values allocated for that gene. The probability of mutation expressed as a percentage or mutation rate ( $P_m$ ) expressed as a ratio determines how often genes on the chromosomes will be flipped. A very high mutation rate widens the search space to an extent that, the probability of convergence to global optima becomes rare. Whereas a low mutation rate drastically shrinks the search space, eventually leading the GA to get stuck in a local optimum. **Lima et al. (2005)** suggested that the mutation rate should not be higher than 30%.

A balance between the exploitation and exploratory abilities of GA is maintained by altering the values of crossover fraction and mutation rate or probability and measuring their cumulative effect on the performance of GA in terms of the fitness function value. A combination of crossover fraction and mutation rate that provides the least value of the fitness function value after a predefined number of generations is adopted for GA aided optimization of ANN.

**d. Stopping criteria for GA**

The GA is initialized using a set of chromosomes population. After that the fitness of each chromosome is evaluated, selection of fitter chromosomes is made, a pair of offspring is generated using crossover operator, and finally, the mutation operator is employed to alter chromosomes to achieve larger exploratory search space. The GA optimization uses the stochastic operators, to proceed from one generation to the next generation, with an overall improvement in the fitness of the chromosome population. These processes are stopped once the GA has reached either the maximum generations allowed for the optimization process or fitness function has achieved a minimum threshold value or no further improvement in fitness value is noticed for a number of consecutive generations.

**2. Applying the LM algorithm for fine tuning of GA evolved weights and biases**

The neural network weights and biases evolved using GA in steps (a) to (d) are assigned to the Levenberg-Marquardt (LM) backpropagation algorithm to, carry forward the optimization of neural network through fine tuning of the initial weights and biases. The LM algorithm uses the forward propagation of information and backpropagation of errors to update the weights and biases systematically so that the errors between actual and predicted outputs are reduced to a minimum threshold value. The validation error is monitored at each iteration cycle to check the overfitting or over training of the neural network. The training is stopped once validation error begins to rise.

**4.4 Modeling slump of RMC using first order and second order regressions**

Regression analysis is a commonly used statistical technique for constructing a quantitative relationship between the dependent variable and one or more independent variables. A regression model is characterized by a mathematical relationship between a single dependent variable or the response  $y$ , that is dependent

on  $k$  independent or regressor variables namely,  $x_1, x_2, x_3, \dots, x_k$  (Montgomery, 2009). In the present study, there are seven independent variables represented by the design mix constituents of RMC and one dependent variable signifying the slump value. These models can be of first order, second order or higher order depending on the degree of polynomial associated with the regression equation.

First order regression models contain a number of explanatory or independent variables that are employed to predict the outcome of the response or dependent variable. A first order regression model is also termed as Multiple Linear Regression (MLR) model since it fits a linear relationship between a dependent or response variable and a number of independent or regressor variables and its generalized form is given by:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \varepsilon \tag{4.8}$$

where,  $\beta_0$  is the intercept,  $\beta_i$  ( $i=1,2,\dots,k$ ) are the partial regression coefficients and  $\varepsilon$  is the random error. If the model contains  $k$  independent variables namely,  $x_1, x_2, \dots, x_k$  (proportions of design mix constituents of RMC) each having unknown coefficients  $\beta_1, \beta_2, \dots, \beta_k$  respectively, then the response of the model (slump value) can be written in the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \tag{4.9}$$

or, if there are  $m$  sets of observations  $x_{1i}, x_{2i}, \dots, x_{ki}$ , then the model for the  $i^{\text{th}}$  observation is:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_k x_{ki} + \varepsilon_i, \quad i = 1, 2, \dots, m \tag{4.10}$$

Alternatively, matrix forms the above equation can be written as:

$$Y = X\beta + \varepsilon \tag{4.11}$$

where,

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1m} & x_{2m} & \cdots & x_{km} \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_m \end{bmatrix} \tag{4.12}$$

The second order regression model is used to fit a curvilinear relationship between the dependent or response variable and a number of independent or regressor variables. This model represents a polynomial equation of second degree and contains the quadratic and the interaction effect of the two independent variables, and its generalized form is:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{j=1}^{i-1} \sum_{i=2}^k \beta_{ji} x_j x_i + \varepsilon \quad (4.13)$$

where,  $y$  is dependent variable or response variable (slump of concrete) and  $\beta_i, \beta_{ii}, \beta_{ji}$  represents the linear, quadratic and interaction effects and  $\beta_0$  is the intercept term. The terms  $x_i, x_j, x_k$  represent the independent variables or influencing variables (proportions of the design mix constituents of RMC).

The coefficients of the regression model defined above can be evaluated using the method of least squares by minimizing the sum of squares of deviations of the observations from true regression line or minimize  $\left( \sum_{i=1}^m \varepsilon_i^2 \right)$ . Using this method the unknown coefficients of the regression line can be computed as:

$$\beta = [X^T X]^{-1} [X^T Y] \quad (4.14)$$

In the case of multiple linear regression of first order the matrix  $X$  contains the values of independent variables  $x_i$ , the matrix  $\beta$  contains the unknown linear coefficients and matrix  $Y$  contain the values of the response variable. Apart from variables mentioned above, in the case of second-order multiple regression models, the matrix  $X$  contains the quadratic terms  $x_i^2$  and the interaction terms  $x_j x_i$  and matrix  $\beta$  contain the unknown quadratic and interaction coefficients  $\beta_{ii}$  and  $\beta_{ji}$  respectively.

#### **4.5 Evaluating performance of the models**

The performance of the concrete slump models discussed in the preceding section, namely GA, LMBNN, GA-LMBNN and first order and second order regression models are required to be assessed to compare their performances and



predictive accuracy. The study uses six different statistical performance metrics namely, root mean square error (RMSE), mean absolute percentage error (MAPE), correlation coefficient (R), coefficient of efficiency (E), root mean square error to the observation's standard deviation ratio (RSR) and percent bias error (PBIAS) given by:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_i - P_i)^2} \quad (4.15)$$

$$MAPE(\%) = \frac{1}{N} \left( \sum_{i=1}^N \frac{|T_i - P_i|}{T_i} \right) \times 100 \quad (4.16)$$

$$R = \frac{\sum_{i=1}^N (T_i - \bar{T})(P_i - \bar{P})}{\sqrt{\sum_{i=1}^N (T_i - \bar{T})^2 \sum_{i=1}^N (P_i - \bar{P})^2}} \quad (4.17)$$

$$E = 1.0 - \frac{\sum_{i=1}^N (T_i - P_i)^2}{\sum_{i=1}^N (T_i - \bar{T})^2} \quad (4.18)$$

$$RSR = \frac{RMSE}{STDEV_{actual}} = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (T_i - P_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (T_i - \bar{T})^2}} \quad (4.19)$$

$$PBIAS(\%) = \frac{\sum_{i=1}^N (P_i - T_i)}{\sum_{i=1}^N T_i} \times 100 \quad (4.20)$$

where,  $T_i$  and  $P_i$  are the target or actual value and ANN predicted value,  $\bar{T}$  and  $\bar{P}$  denote the mean of actual and ANN predicted values, respectively.  $N$  are the number of data pairs used for the study.

The root mean square error (RMSE) is one of the most important and widely used performance metrics for judging the prediction accuracy of the ANN model. It compares the observed or the target value with the values predicted by the model and computes the square root of the average of the squared residual error. RMSE measures the prediction accuracy of the model in terms of variance and degree of bias and therefore because of its quadratic nature, this error index is biased towards large errors

or outliers (**Willmott and Matsuura, 2005**). The lower the RMSE, the better is the prediction accuracy of the model. The mean absolute percentage error (MAPE) is a dimensionless error statistic, which compares the model's errors to the actual data and presents them in the form of a percentage. MAPE provides an intuitive way of estimating the importance of the errors, but may provide meaningless comparisons among series if the actual or observed value in the denominator is significantly smaller than the error value in the numerator. This problem arises when there are numerous outliers in a series. MAPE statistic ranges from 0% to 100%, a lower value indicating good model prediction and vice versa.

The Pearson's correlation coefficient (R) is a statistical measure that quantifies the relationship between the two variables. Since the correlation coefficient (R) indicates the strength of two linearly associated variables, it does not yield the desired inferences when the relationship is not linear. The value of the correlation coefficient is over sensitive to extreme value outliers in comparison to the values near the mean (**Legates and Davis, 1997**), as they significantly influence the slope of the regression line, but is insensitive to additive and proportional errors (**Legates and McCabe, 1999**). The value of correlation coefficient (R) varies in the range -1 to +1. The value of R close to +1 indicates that the model has achieved better prediction accuracy with a high positive linear association between the actual or target value and the model's predicted value.

The coefficient of efficiency E or Nash-Sutcliffe efficiency (**Nash and Sutcliffe, 1970**) is a normalized statistic that provides a ratio of two variances namely, residual error variance and variance in actual or target data. The value of this statistic ranges from  $-\infty$  to +1. The value of E close to +1 indicates a close agreement between the actual and predicted data and their alignment with respect to the 1:1 line. The value of  $E < 0$  indicates the unacceptable performance of the model. The E statistic is an improvement over correlation coefficient (R), as the value takes into account the differences in the means and variances of the actual and model-predicted values, but still E is sensitive to the outlier values.

The root mean square error to the observations standard deviation ratio (RSR) developed by **Moriasi et al. (2007)**, standardizes root mean square error (RMSE) by incorporating both an error index and the standard deviation of the actual data. If the value of RMSE is 0 then, RSR statistic achieves an optimal value of 0

indicating a perfect model prediction. Hence, a lower value of RSR indicates a better agreement between actual and predicted values. Percent bias (PBIAS) statistic measures the average tendency of predicted values to be larger or smaller than their corresponding actual or target values (Gupta *et al.*, 1999). This statistic is also a measure of the model's ability to predict a value situated apart from the mean value. The optimal value of PBIAS is 0.0, which indicates a perfect model prediction. A positive PBIAS expressed as a percentage indicates over-prediction of the model, whereas a negative value indicates model's under-prediction (Srinivasulu and Jain, 2006).

As discussed above, a single performance metric cannot provide an unbiased estimate of model's prediction ability. Hence, a combined of the above statistical performance metrics has been used in the study to derive holistic inferences regarding the prediction accuracy of the slump models. The performance of each slump model is evaluated using the statistical performance metrics. The model showing the best performance is selected for further analysis.

#### **4.6 Sensitivity analysis**

Sensitivity analysis is an important means of drawing logical inferences regarding a highly complex and non-linear mathematical model, without having to solve the complex problem. The sensitivity analysis provides the response of the model subjected to parametric variations, allowing to understand the internal mechanism of the problem and to determine the key variables influencing a system model. The sensitivity analysis in the present study deals with the evaluation of the importance of each concrete design mix constituent on the slump value and deduce the response on the slump value brought about by varying the proportions of each design mix constituent.

##### **4.6.1 Variable importance using Connection Weights method**

In the case of regression models dealing with a few variables, it is easy to interpret the impact of the independent variables on the dependent variables. Normally, the problems encountered in real life are not simple enough to be solved

using conventional mathematical techniques namely, linear or non-linear regression models. The use of ANN methodology for solving real life problems provides the user with the liberty of incorporating a number of inputs or independent variables influencing a particular phenomenon. Moreover, in contrast to the conventional mathematical modeling techniques requiring prior knowledge about the interactions between independent and dependent variables, the ANN's adaptive nature do not require such prior knowledge, as they can build the underlying interactions based on the input-output patterns presented during their learning phase. These characteristics of ANN categorize them as "black boxes". This is attributed to the internal structure of the trained neural network comprising a set of numbers that makes it difficult to relate back to the problem in a meaningful fashion (**Paliwal and Kumar, 2011**).

ANNs are in vogue due to their ability to approximate almost any functional relationship between the input-output data pairs. However, without interpretation of the relative importance of the parameters in the system, the utility of the ANN is limited (**Kemp et al., 2007**). Hence, a trained neural network model that provides answers to interactions between the input and output parameters is therefore of prime importance. This requirement is emphasized in the case of modeling concrete slump that, exhibits a highly complex functional interaction between the concrete's design mix constituents and the slump value. The composite nature of concrete does not allow the direct determination of the importance of its design mix constituents on the slump value. However, the neural network weights that form the links between the inputs and outputs of the ANN model also act as links between problem and solution (**Olden and Jackson, 2002**) and carry meaningful information about the problem within the trained neural network (**Kalogirou, 2000**). The neural network weights bear similarity to parameter coefficients of a standard regression model, and their strength and nature dictate the way information is processed by the neural network. A positive and larger value of weight increases the relative influence and association of input variable with the output variable. On the other hand, a negative or smaller value of weight suppresses this influence. The neural network weights thus provide a meaningful interpretation of the effect of input variables and help in understanding the so-called "black box" nature of ANN (**Acciani et al., 2006**).

In the present study, the connection weight method proposed by **Olden et al. (2004)** has been used to quantify the importance of design mix

constituents of concrete on the slump value. In this method, the product of the connection weights between the input node and hidden nodes with the connection weights from hidden node to output nodes is summed up for all input nodes and is defined as the relative importance of the input variable given by **Equation 4.21**.

$$RI_x = \sum_{y=1}^h (w_{xy} \times w_{yz}) \quad (4.21)$$

where,  $RI_x$  is the relative importance of input neuron or input variable  $x$ ,  $y$  is the total number of hidden layer neurons,  $z$  is the total number of output neurons,  $w_{xy}$  is the weight of the connection from input neuron  $x$  to hidden layer neuron  $y$  and  $w_{yz}$  is the weight of the connection from hidden layer neuron  $y$  to output layer neuron  $z$ . The larger the sum of the product of weights  $w_{xy}$  and  $w_{yz}$  for a particular input neuron, more is the relative importance ( $RI_x$ ) asserted by that input neuron  $x$  on the output neuron  $z$ .

#### 4.6.2 Response trace plots

The response trace is a plot of the response of the variable as one move away from a standard design mix. The design mix proportions of concrete are denoted as weight in kilogram per cubic meter of concrete. Hence, the sum of the volume of each constituent is  $1 \text{ m}^3$ , which can be denoted as:

$$\frac{W_{cement}}{\rho_{cement}} + \frac{W_{flyash}}{\rho_{flyash}} + \frac{W_{sand}}{\rho_{sand}} + \frac{W_{aggregate}}{\rho_{aggregate}} + \frac{W_{superplasticizer}}{\rho_{superplasticizer}} + \frac{W_{water}}{\rho_{water}} = 1 \quad (4.22)$$

where,  $W$  is the weight of each design mix constituent in kg and  $\rho$  is the material density of each constituent in  $\text{kg/m}^3$ . If any one of the constituents is varied, the change will be reflected on other constituents. If  $r_i$  is the current volumetric proportion of a constituent and  $\Delta_i$  is the volumetric change brought about in the constituent then, the new the volumetric proportion of this constituent is (**Myers et al., 2009**):

$$x_i = r_i + \Delta_i \quad (4.23)$$

To keep the total volume equal to 1 cubic meter, the proportions of other constituents are changed as:

$$x_j = r_j - \frac{r_j}{1-r_i} \Delta_i; i \neq j \quad (4.24)$$

The revised proportions of the concrete mix are computed using the above method and the selected model is used to predict the response or slump based on the revised proportions. The slump value is then plotted along the y-axis, and the variable concrete design mix constituent is plotted along the x-axis to provide the response plot for that constituent.

#### **4.7 Assessing applicability of the developed slump model for a different RMC batching plant**

A total of 100 mix proportions of the concrete constituents namely, cement, PFA, sand, CA (20mm), CA (10mm), superplasticizer and water and corresponding slump value were collected from a second RMC batching plant located in the vicinity of the first RMC batching plant. The details and statistics of the data are exhibited in **Table 4.3**.

**Table 4.3:** Details and statistics of RMC data used for validating the applicability of the developed model for concrete slump

RMC data	Statistics				Specific Gravity
	Min.	Max.	Mean	SD	
Cement (kg/m <sup>3</sup> )	120	450	271.5900	73.1666	3.15
Pulverized fuel ash (kg/m <sup>3</sup> )	0	200	59.4400	52.1753	2.10
Sand (kg/m <sup>3</sup> )	592	900	766.9600	52.9335	2.67
Coarse aggregate 20mm (kg/m <sup>3</sup> )	438	760	574.0000	51.3496	2.65
Coarse aggregate 10mm (kg/m <sup>3</sup> )	348	600	534.0800	44.9728	2.65
Superplasticizer (kg/m <sup>3</sup> )	1.70	5.50	3.5510	0.6950	1.15
Water (kg/m <sup>3</sup> )	140	189	166.4900	8.6146	1.00
Slump (mm)	100	170	156.7000	8.0785	-

The selected model for concrete slump is harnessed to predict the slump for the design mix proportions derived from the second RMC plant. This has been explicitly done to ascertain the whether the slump model developed for a particular RMC plant can be reliably and robustly applied for assessment of the concrete slump for design mix proportions of a different RMC plant.

#### **4.8 Developing decision support tool for estimating concrete slump**

The trial mix procedure for designing a concrete mix of customized workability involves tedious experimental procedures demanding time and resources. To facilitate quick determination of slump value for the concrete design mix, the study attempts to harness the best performing model for concrete slump for developing a decision support tool, which will provide a fair assessment of slump value for proportions of concrete design mix constituents, without performing actual slump tests at the batching plant. The tool is developed using the functionalities of MSEXcel and MATLAB software. MSEXcel is used as front end and MATLAB is used as backhand software. The neural network based model for the slump is loaded in MATLAB and connected to MSEXcel using Excel link provided in the MATLAB.

#### **4.9 Summary**

The data for the research were collected from two different RMC batching plants. The data constituted the design mix proportions of concrete constituents namely, cement, pulverized fuel ash, sand, coarse aggregate (20mm), coarse aggregate (10mm), superplasticizer and water in  $\text{kg/m}^3$ , along with their corresponding slump value in mm. The Neural Network Toolbox and Global Optimization Toolbox included in the commercially available software MATLAB 7.1 (R14 SP3) (Version 7.1.0.246) were used to implement the BPNN and GA respectively.

The data collected from first RMC batching plant were used for modeling the concrete slump. The entire data are divided into three statistically similar and representative subsets namely, training, validation and test datasets such that, their statistics namely, mean and standard deviation are marginally different from that of

the population. Before training the data are normalized to minimize the bias within the network towards a particular feature. The topology for a MFNN is represented by the number of layers, number of neurons in each layer and the type of transfer or activation function. In the study three layer MFNN has been used with seven input layer neurons representing the design mix proportions of the RMC constituents namely, cement, pulverized fly ash (PFA) or fly ash, sand, coarse aggregate (CA) 20mm, coarse aggregate (CA) 10mm, superplasticizer (SP) and water and one output layer neuron representing the slump value of concrete. The systematic updating of the neural network weights and biases for MFNN is accomplished by the Levenberg-Marquardt (LM) backpropagation training algorithm. This neural network is hereafter called the Levenberg-Marquardt backpropagation neural network (LMBNN). A tangent hyperbolic transfer function known for speeding up the training process has been used in the hidden layer, and a linear transfer function has been used for output layer to facilitate comparison of actual and predicted slump values.

The average standard deviations of errors (AVSD) and average mean absolute error (AVMAE) for both training and validation datasets for varying number of hidden neurons in the range 4 to 15 were examined. The neural network topology that provided the least validation AVSD and AVMAE is chosen for further analysis. The hybrid methodology (GA-LMBNN) for modeling slump of RMC presented in the study comprise of two stages. The first stage utilizes the global optimization and search of GA for evolving the optimized initial weights and biases for ANN. For this purpose, the weights and biases for the selected neural network architecture are coded as genes of the chromosomes. In the second stage, the LM algorithm starts with the GA evolved optimized set of weights and biases and iteratively updates and fine tunes these weights and biases to reach a global optimum. The conventional LMBNN model, on the other hand, was initialized with random values of weights and biases in the range [-0.5, 0.5]. Apart from neural network models, the slump of concrete was modeled using the first order and second order regression models.

The study uses six different statistical performance metrics namely, root mean square error (RMSE), mean absolute percentage error (MAPE), correlation coefficient (R), coefficient of efficiency (E), root mean square error to the observation's standard deviation ratio (RSR) and percent bias error (PBIAS) to compare the effectiveness and prediction accuracy of the developed models. The



sensitivity analysis in the present study deals with the evaluation of the importance of each concrete design mix constituent on the slump value using connection weights method and deduce the response on the slump value brought about by varying the proportions of each design mix constituent using the response trace plots. The robustness and universal applicability of the slump model are assessed by, utilizing it to predict the slump for design mix data procured from the second RMC batching plant. For easing the burden of performing regular multiple trials with different design mix proportions, a decision support tool using the selected model is developed for estimating the slump at the RMC batching plant.

## **Chapter 5**

### **Results and Discussion**

---

- **Introduction**
  - **Determining the optimal number of neurons in the hidden layer**
  - **Genetic algorithms (GA) assisted training of artificial neural networks**
  - **Concrete slump models**
  - **Neural networks models versus regression models**
  - **Relative importance of RMC design mix constituents on the slump value**
  - **Response trace plots**
  - **Performance and reliability of the concrete slump model for different RMC batching plant**
  - **Decision support tool to estimate slump for the RMC design mix proportions**
  - **Summary**
-

## 5.1 Introduction

The previous chapter has comprehensively dealt with the methodology adopted in the study. This chapter presents the results of the study and further discusses their outcome. The chapter has been divided into sections that deal with determination of number of hidden layer neurons and parameters of GA, evolving neural network weights and biases using GA, training, validation and testing of artificial neural network using GA evolved initial weights and biases, comparison of slump models namely, GA-LMBNN, LMBNN, first order and second order regression models, evaluating the relative importance of concrete design mix constituents on slump value, plotting response trace plot of slump value for each concrete design mix constituent, assessing the applicability of the trained hybrid GA-LMBNN model for predicting concrete slump based on the concrete design mix data collected from a different RMC batching plant and developing decision support tool for predicting slump of RMC based on design mix constituents.

## 5.2 Determining the optimal number of neurons in the hidden layer

The concrete's design mix proportion data collected from first RMC batching plant is used for developing the neural network model. The architecture of Multilayer Feedforward Neural Networks (MFNN) primarily comprises of three layers namely, an input layer, a hidden layer, and an output layer. For the present study, the input layer comprised of seven neurons namely, cement, pulverized fuel ash (PFA), sand, coarse aggregate (CA) 20mm, coarse aggregate (CA) 10mm, superplasticizer (SP) and water. The output layer comprised of a single neuron representing the slump of concrete. A certain degree of non-linearity in the neural network architecture is introduced by using hyperbolic tangent sigmoid transfer function (*Tansig*) for the hidden layer neurons. The comparison of actual and neural network predicted slump value is facilitated by incorporating the linear transfer function (*Purelin*) for the output neuron.

The number of hidden layer govern the complexity of the MFNN and are needed in sufficient number to map the complex functional relationship between input-output data patterns and enable efficient learning and generalization of the neural network model. A trial and error approach is employed for determining the

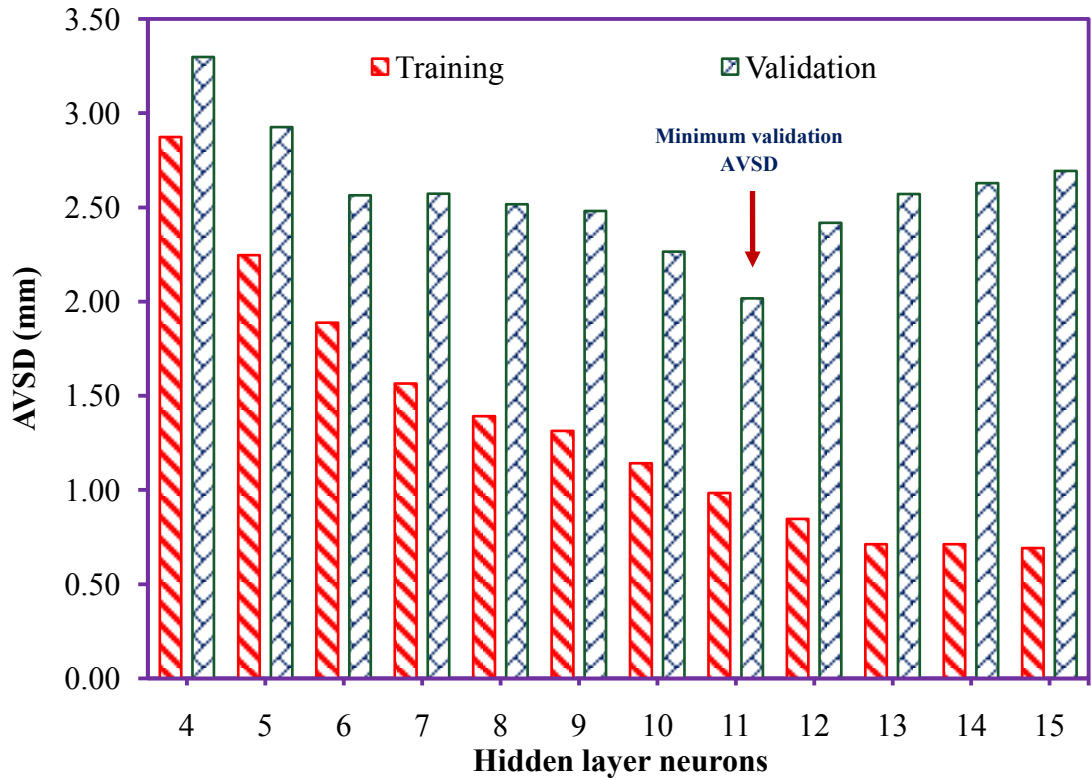
optimal number of hidden layer neurons. The neurons in the hidden layer were varied in the range [4, 15] and the neural network model was trained for 1000 training cycles or epochs using the training data set comprising of 346 datasets. The training of the MFNN model using Levenberg-Marquardt backpropagation algorithm (LMBNN) was repeated ten times using a different set of randomly initialized weights and biases.

A major problem with neural networks training is the loss in generalization ability with increase in the network complexity. This is primarily due to over-fitting or over learning of training data. The issue of over-fitting is addressed by presenting the neural network model with the validation dataset comprising of 74 datasets and monitoring the validation error at each training epoch. The standard deviation and mean absolute of training and validation errors was recorded at each increment of hidden layer neuron. The over-fitting of the neural network model was minimized by early stopping of the neural network training, if the standard deviation and mean absolute of validation errors begins to rise with an increase in number of hidden layer neurons. The numbers of hidden layer neurons were varied and the standard deviation and mean absolute of training and validation error was recorded and averaged for ten independent runs of each model. The average standard deviation of errors (AVSD) and average mean absolute error (AVMAE) for different neural network configurations is exhibited in **Table 5.1**. The AVSD and AVMAE plotted against hidden layer neurons are shown in **Figure 5.1** and **Figure 5.2** respectively.

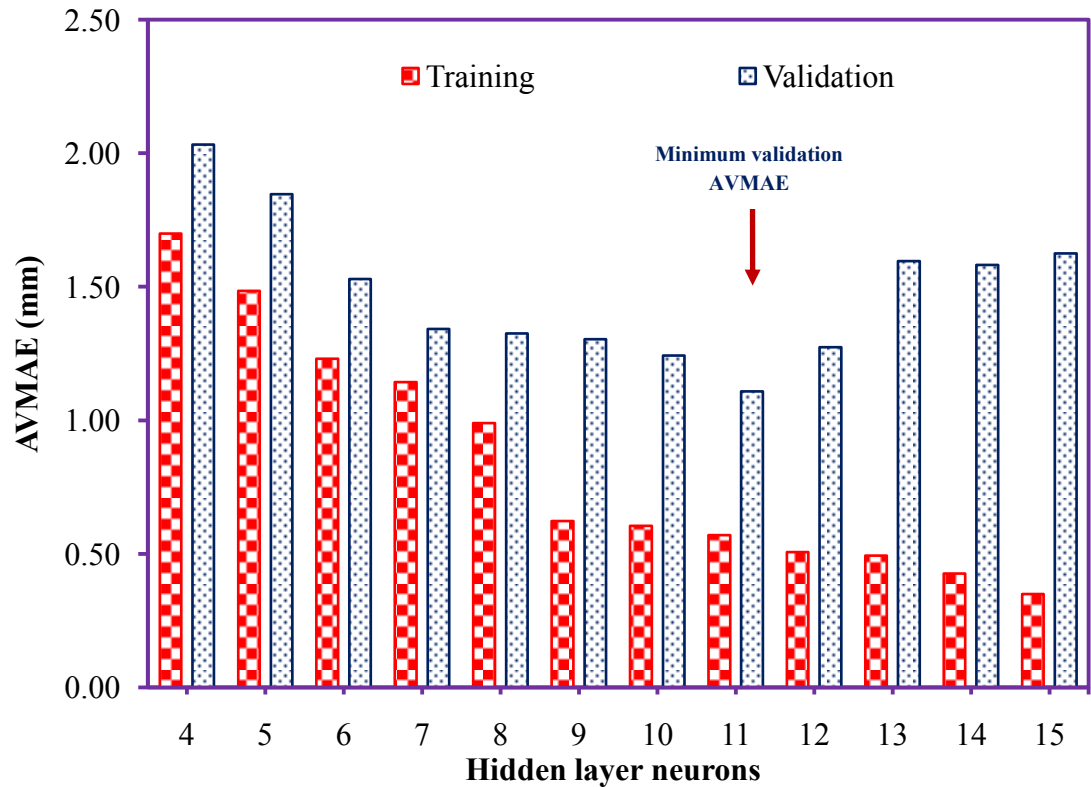
It is seen that, with an increase in the number of hidden layer neurons, the AVSD and AVMAE value gradually goes on decreasing from 2.8737 mm and 1.7000 mm respectively at four hidden layer neurons to 0.6913 mm and 0.3498 mm respectively at fifteen hidden layer neurons. However, the validation error first decreases and then increases, attaining a minimum average value of standard deviation and mean absolute error 2.0174 mm and 1.1088 mm respectively at eleven hidden layer neurons, indicating over-fitting of the neural network with an increase in the hidden layer neurons. It shows that, the neural network architecture comprising of eleven hidden layer neurons yielded the best generalization. The neural network architecture for the present study thus constitute three layers of neurons, namely an input layer having seven neurons, a hidden layer having eleven neurons and an output layer having one neuron and is represented by 7-11-1. The schematic of the optimal neural network architecture is shown in **Figure 5.3**.

**Table 5.1:** Training and validation AVSD and AVMAE for different neural network configurations

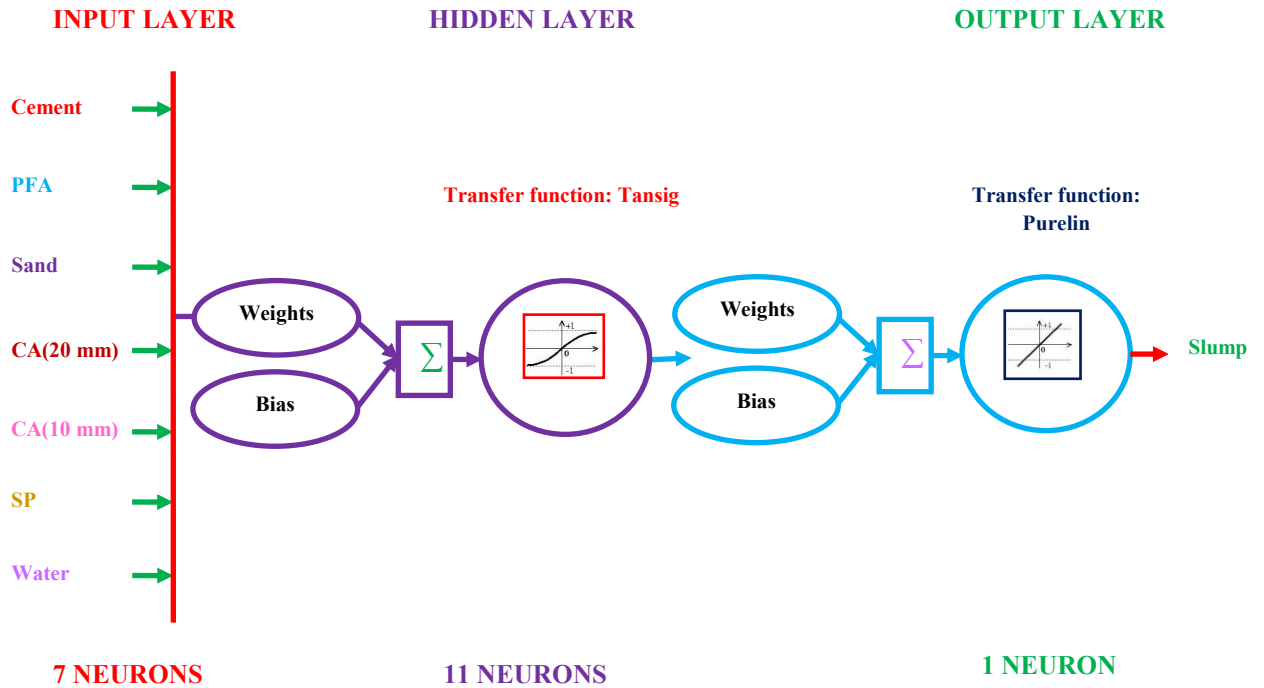
Neural network architecture			Transfer function		AVSD (mm)		AVMAE (mm)	
Input layer	Hidden layer	Output layer	Hidden layer	Output layer	Training	Validation	Training	Validation
7	4	1	<i>Tansig</i>	<i>Purelin</i>	2.8737	3.2980	1.7000	2.0334
7	5	1	<i>Tansig</i>	<i>Purelin</i>	2.2472	2.9264	1.4850	1.8475
7	6	1	<i>Tansig</i>	<i>Purelin</i>	1.8885	2.5639	1.2307	1.5297
7	7	1	<i>Tansig</i>	<i>Purelin</i>	1.5648	2.5737	1.1442	1.3424
7	8	1	<i>Tansig</i>	<i>Purelin</i>	1.3919	2.5171	0.9908	1.3261
7	9	1	<i>Tansig</i>	<i>Purelin</i>	1.3136	2.4807	0.6233	1.3037
7	10	1	<i>Tansig</i>	<i>Purelin</i>	1.1415	2.2659	0.6058	1.2427
<b>7</b>	<b>11</b>	<b>1</b>	<b><i>Tansig</i></b>	<b><i>Purelin</i></b>	<b>0.9834</b>	<b>2.0174</b>	<b>0.5703</b>	<b>1.1088</b>
7	12	1	<i>Tansig</i>	<i>Purelin</i>	0.8463	2.4183	0.5074	1.2746
7	13	1	<i>Tansig</i>	<i>Purelin</i>	0.7123	2.5719	0.4940	1.5972
7	14	1	<i>Tansig</i>	<i>Purelin</i>	0.7120	2.6291	0.4272	1.5818
7	15	1	<i>Tansig</i>	<i>Purelin</i>	0.6913	2.6929	0.3498	1.6259



**Figure 5.1:** Variation of average standard deviation (AVSD) of training and validation error with hidden layer neurons.



**Figure 5.2:** Variation of average mean absolute (AVMAE) of training and validation error with hidden layer neurons.



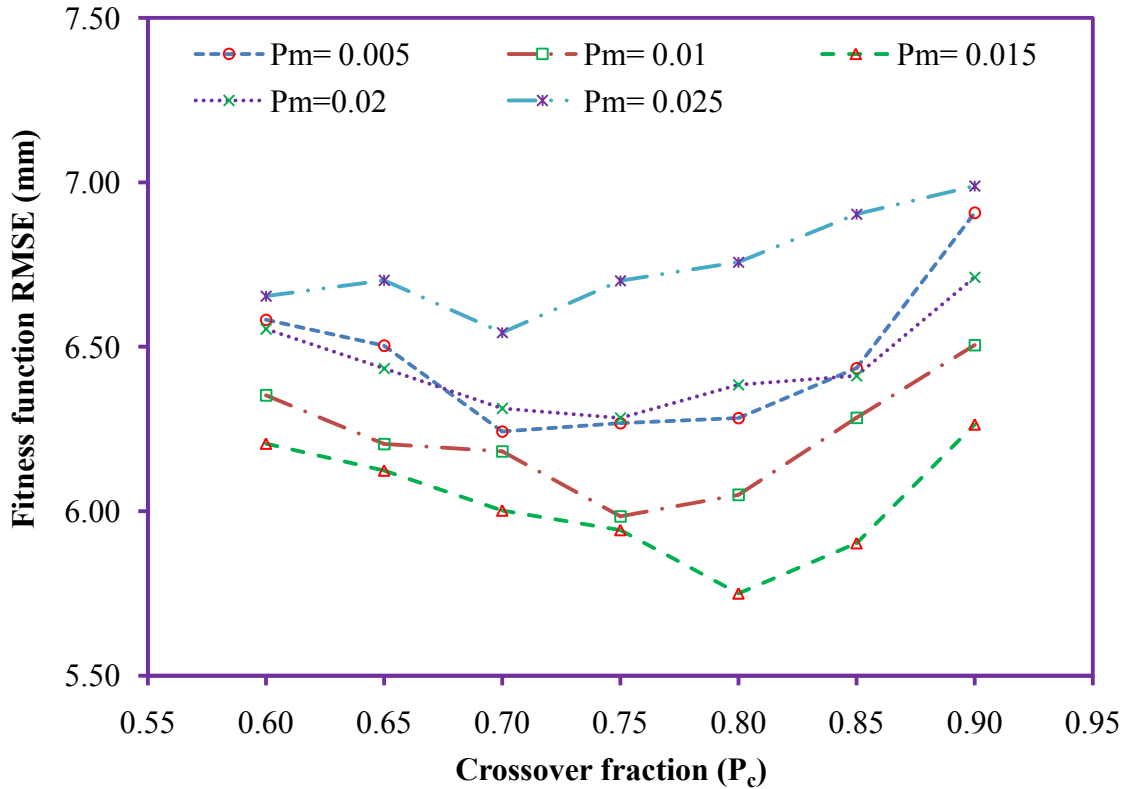
**Figure 5.3:** Schematic of the optimal neural network architecture used in the study

### 5.3 Genetic algorithms (GA) assisted training of artificial neural networks

#### 5.3.1 Estimating optimal GA parameters

In the present study, the selected MFNN configuration (7-11-1) has 100 weights and biases. At the start of the GA optimization, the weights and biases are initialized as genes of the chromosomes. The global search of GA updates the weights and biases of MFNN to reduce the value of the fitness function (RMSE) gradually over a number of generations till stopping criteria is achieved. The basic parameters of GA namely, population size, crossover fraction and mutation rate that influence the performance of the GA are required to be assessed and fixed for achieving a global optimum through a balanced exploration and exploitation of search space. Since GA is a stochastic search tool, hence, the procedure of determining the optimal values of population size, crossover fraction  $P_c$ , and mutation rate  $P_m$  was repeated fifty times and the average value of the best fitness function is determined. The other parameter of GA namely, the maximum number of generations and stall generation limit were fixed at 200 and five respectively. Keeping population size equal to 30, the crossover fraction and mutation rate were varied in the range [0.6, 0.9] and [0.005, 0.025]

respectively. The average values of the best fitness function (RMSE) plotted for different combinations of  $P_c$  and  $P_m$  are shown in **Figure 5.4**. It is seen that minimum value of the fitness function is achieved by adopting crossover fraction  $P_c$  as 0.80 and mutation rate  $P_m$  as 0.015.



**Figure 5.4:** Fitness function with varying crossover fraction ( $P_c$ ) and mutation rate ( $P_m$ )

Once the optimal values of crossover fraction and mutation rate have been determined, the population size is varied in the range [20, 80]. The values of the fitness function (RMSE) and CPU time averaged over fifty runs of GA exhibited in **Table 5.2** show that, as the population size is increased, the improvement in fitness function reduces and at higher population size, it becomes almost negligible. An increase in the population size leads to more function evaluations, increasing the CPU time taken by GA. However, it is seen that the increase in CPU time is significant in comparison to the improvement in the fitness function achieved. Hence, to strike a balance between the performance and computational time taken by GA, an initial population size of 40 chromosomes is adopted in the study. **Table 5.3** shows the optimal GA parameters used for the study.



**Table 5.2:** CPU time and value of fitness function for different GA population sizes

Population Size	Fitness function value (mm)	CPU time (sec)	Improvement in fitness (mm)	Increase in CPU time (sec)
20	5.9525	6.7956	-	-
30	5.7417	9.7939	0.2108	2.9983
40	5.5918	13.5106	0.1499	3.7167
50	5.5312	23.4561	0.0606	9.9455
60	5.5024	38.6110	0.0288	15.1549
70	5.4926	62.2350	0.0098	23.6240
80	5.4888	94.4752	0.0038	32.2402

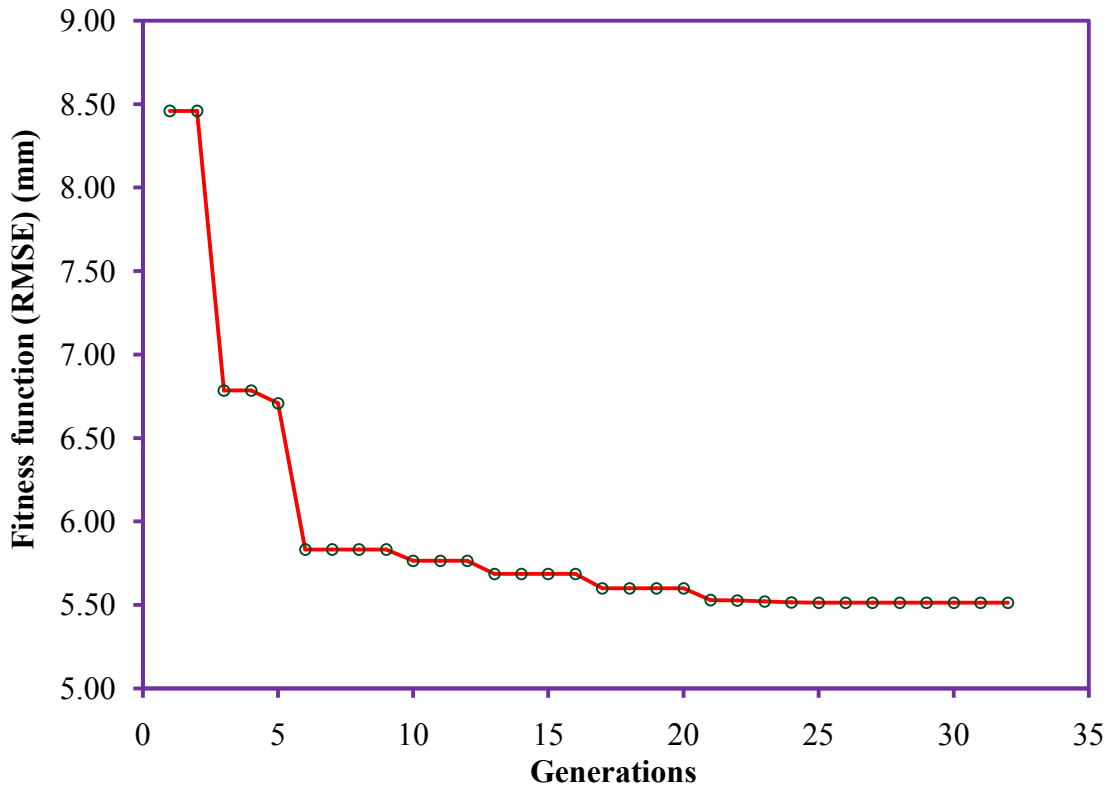
**Table 5.3:** Optimum GA parameters used for the study

GA parameter	Values
Initial Population Size	40
Fitness function	RMSE
Genetic operators	Scattered crossover (0.80), Uniform mutation (0.015)
Selection method	Tournament
Generations	200
Max. Stall generations	5

### 5.3.2 Evolving initial neural network weights and biases using GA

The optimization of neural network weights and biases is achieved by GA assisted training of the selected MFNN model having architecture 7-11-1. The MFNN trained using GA is hereafter referred to as GA model. The optimization procedure

adopted by GA attempts to update neural network weights and biases to minimize the value fitness function (RMSE) over a number of population generations. A typical plot of fitness function (RMSE) versus generations is exhibited in **Figure 5.5**. The value of the fitness function is shown to reduce over a number of generations till stalling of the fitness function is noticed over a number of generations. At this point, the algorithm is stopped, and the weights and biases corresponding to the best value of the fitness function are saved.



**Figure 5.5:** Typical plot of fitness function versus population generations for GA

Since GA is a stochastic global search technique, it has a tendency to provide different solutions to the problem during each re-run of the optimization process. Hence, the robustness of the hybrid approach is evaluated by independently running the GA optimization procedure twenty times. The weights and biases for each independent run of GA were recorded. A brief summary of each independent run of GA model, showing the maximum number of generations, the value of best fitness function value achieved and CPU time is exhibited in **Table 5.4**. The average CPU time and population generations taken by GA model during the independent model runs for evolving the optimal set of neural network weights and biases is evaluated as 13.7189 seconds and 39 respectively.

**Table 5.4:** Generations, best fitness function value and CPU time for independent runs of GA

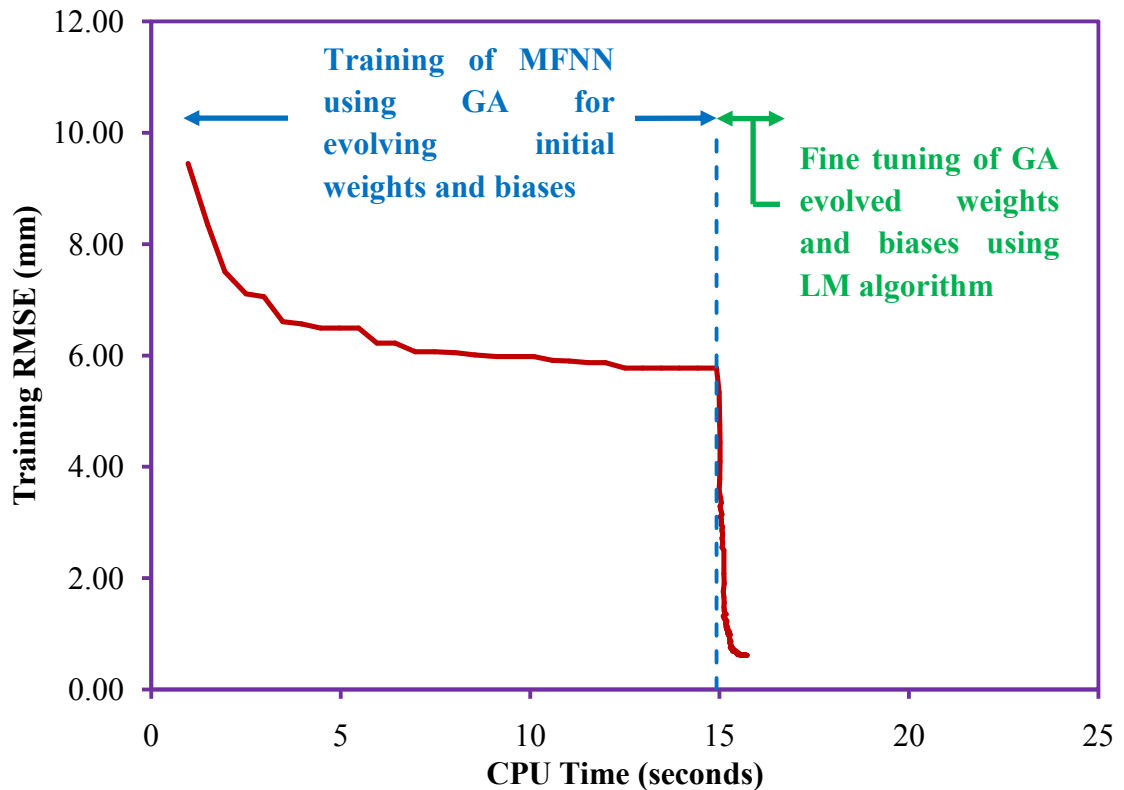
Model run	Gen.	Best fitness function (mm)	CPU time (sec)	Model run	Gen.	Best fitness function (mm)	CPU time (sec)
1	41	5.8549	14.6416	11	33	5.5374	11.9085
2	34	5.5384	12.1533	12	16	6.0434	5.4600
3	50	5.2959	17.6761	13	24	5.7634	8.6959
4	23	6.0492	8.3514	14	59	5.4133	20.7428
5	55	5.3948	19.3816	15	37	5.4484	13.1848
6	39	5.5700	13.8740	16	26	5.6702	9.3840
7	28	5.9670	10.1047	17	57	5.4467	20.0544
8	29	5.6998	10.4177	18	21	5.9853	7.6637
9	33	5.5043	11.8087	19	59	5.3637	21.1428
10	44	5.3303	15.5964	20	63	5.5994	22.1350

### 5.3.3 Training of LMBNN model using GA evolved weights and biases

The twenty independent runs of GA yielded twenty different sets of optimal initial neural network weights and biases. These GA evolved weights and biases were used for initializing the neural network model trained using Levenberg-Marquardt backpropagation algorithm (LMBNN). The hybrid GA-LMBNN model was trained using the training data set and the initial weights and biases were updated using Levenberg-Marquardt (LM) backpropagation algorithm. A typical plot showing the variation of fitness function with CPU time is exhibited at **Figure 5.6**. At the initial stage, the GA is shown to efficiently train the MFNN. However, as training progresses and the search space is narrowed down, the convergence is adversely affected due to GA's weak local search ability. However, at this stage, the global search rendered by GA has evolved the optimal set of initial weights and biases for LM algorithm.

The second stage of training of the hybrid GA-LMBNN is accomplished by fine tuning of the GA evolved optimal initial weights and biases through LM algorithm. The LM algorithm starts from the weights and biases located near the optimal solution and using an efficient local search, update the weights and biases to

converge quickly to an optimum value. A very sharp decrease in RMSE value in a minuscule CPU time is noticeable which is attributed to the LM optimization algorithm.



**Figure 5.6:** Typical plot of fitness function versus CPU time during training of GA-LMBNN model

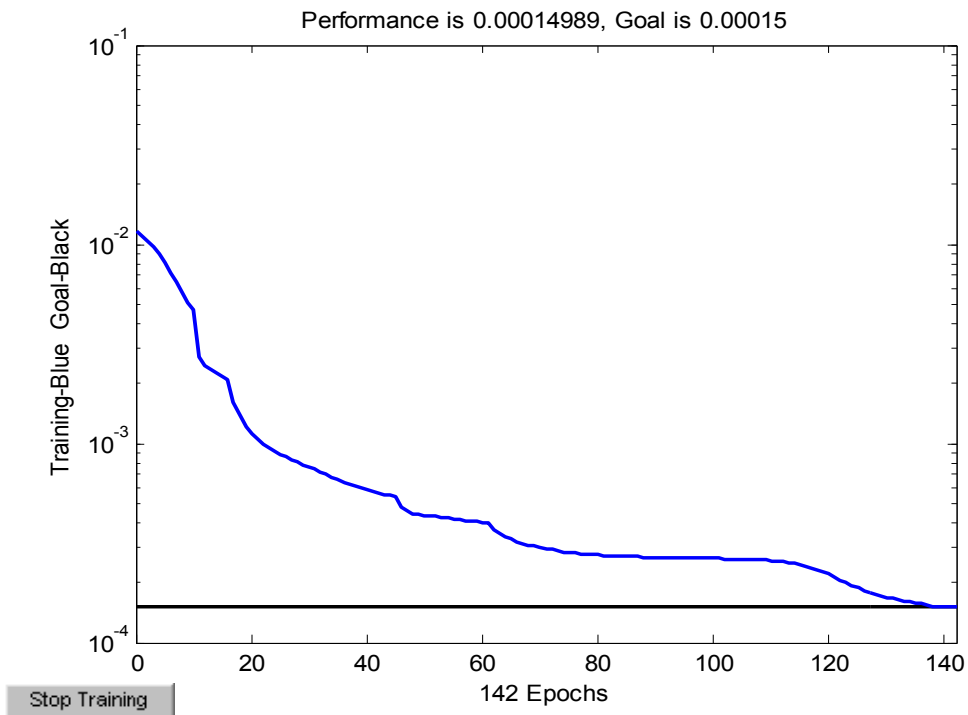
## 5.4 Concrete slump models

### 5.4.1 Neural network models

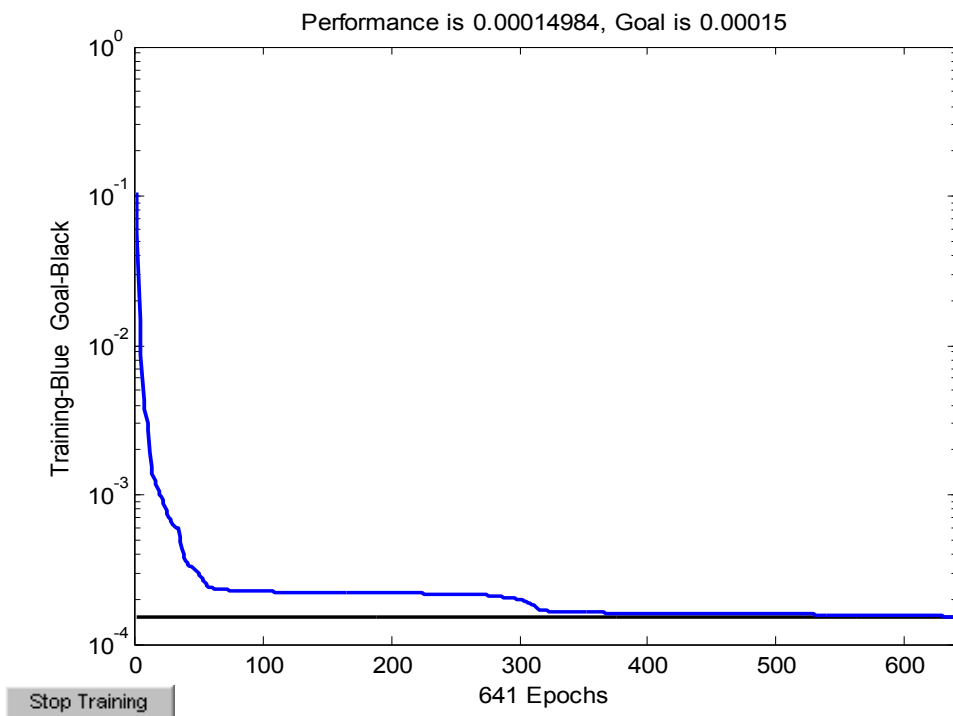
#### 5.4.1.1 Convergence of hybrid GA-LMBNN and conventional LMBNN models

The effectiveness of hybridizing GA with LMBNN model is assessed by comparing its convergence speed during training with the conventional LMBNN model. The LMBNN model was initialized with random values of weights, and biases in the range  $[-0.5, 0.5]$  whereas, the hybrid GA-LMBNN model was initialized using GA optimized weights and biases. The performance goal (normalized MSE) during training for both neural network models was fixed at 0.00015. The training was continued till performance goal was met. A typical training plot of the GA-LMBNN and LMBNN models exhibited in **Figure 5.7** and **Figure 5.8** respectively, reveals that

in comparison to conventional LMBNN model, the hybrid GA-LMBNN approaches the designated performance goal in less number of training epochs or training cycles.

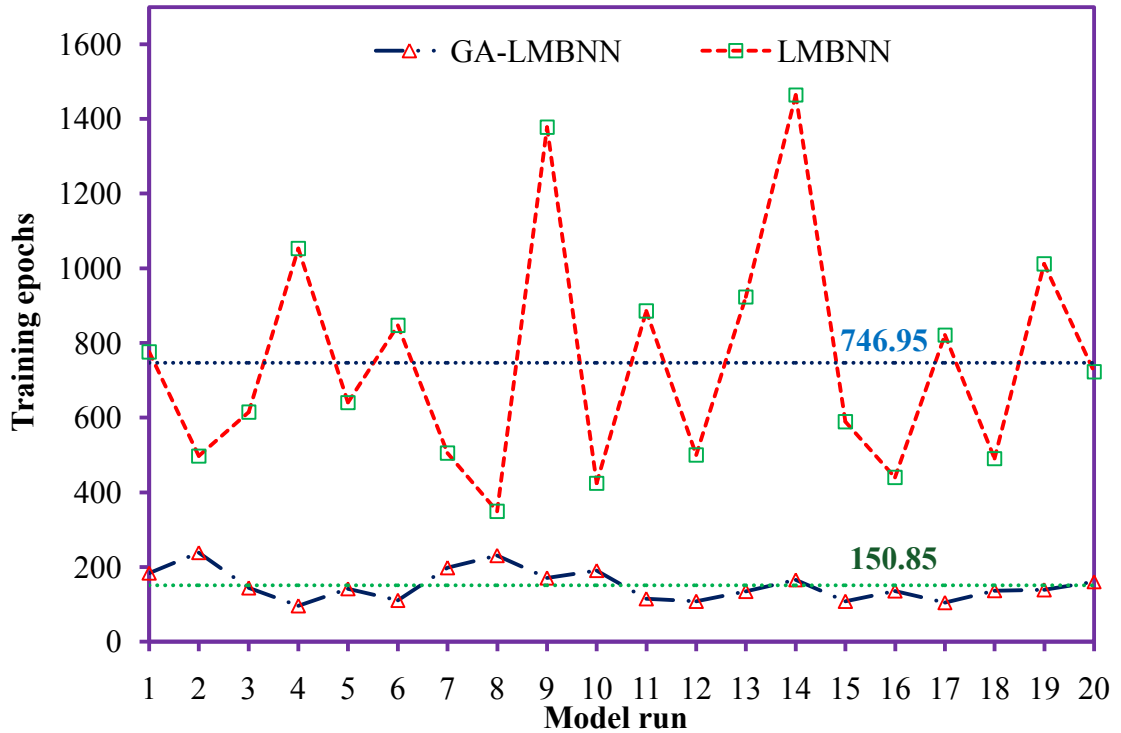


**Figure 5.7:** Typical training plot for GA-LMBNN model

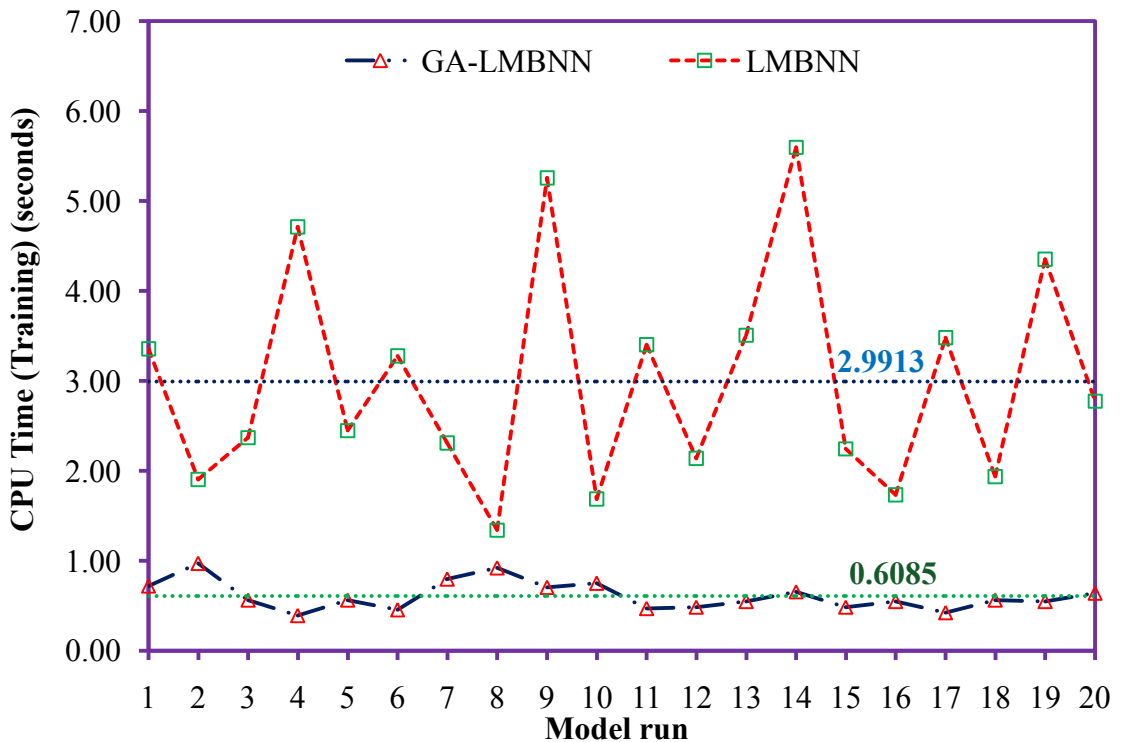


**Figure 5.8:** Typical training plot for LMBNN model

The number of training epochs and CPU time for each independent run of the model were recorded and plotted (**Figure 5.9** and **Figure 5.10**) and their statistics are shown in **Table 5.5**.



**Figure 5.9:** Training epochs for each independent run of the model



**Figure 5.10:** Training CPU time for each independent run of the model

**Table 5.5:** Statistics of training epochs and CPU time for independent runs of LMBNN and hybrid GA-LMBNN models

Statistics	Model			
	LMBNN		GA-LMBNN	
	Epochs	CPU time	Epochs	CPU time
Min.	349	1.3416	96	0.3900
Max.	1465	5.6004	239	0.9687
Mean	746.9500	2.9913	150.8500	0.6085
SD	307.9909	1.2232	41.3283	0.1593

The average number of training epochs and CPU time taken by the hybrid GA-LMBNN model during their independent runs was 150.8500 and 0.6085 seconds respectively. Compared to the hybrid model, the conventional LMBNN model took an average 746.9500 epochs and 2.9913 seconds to reach the same performance goal indicating that, the hybridization methodology leads to an approximate 80% reduction in training epochs and CPU time. The hybrid model is shown to provide lower minimum and maximum values of training epochs and CPU time along with lower standard deviation value of 41.3283 epochs for training epochs and 0.1593 seconds for CPU time as compared to 307.9909 epochs and 1.2232 seconds respectively achieved by the LMBNN model. The results indicate consistency, stability and faster convergence of the hybrid GA-LMBNN model during the training phase, compared to the conventional LMBNN model.

#### 5.4.1.2 Training, validation, and testing performance

An unbiased performance of the slump models developed for the study, namely GA, hybrid GA-LMBNN and conventional LMBNN model, is assessed using six different statistical performance metrics namely, RMSE, MAPE, R, E, RSR and PBIAS. The statistics namely, minimum, maximum, mean and standard deviation of training and validation performance metrics namely, RMSE, MAPE, R, E, RSR and PBIAS were evaluated for twenty independent runs of GA, hybrid GA-LMBNN and conventional LMBNN model. The statistics of training and validation performance metrics are exhibited in **Table 5.6**.

**Table 5.6:** Statistics of the performance metrics

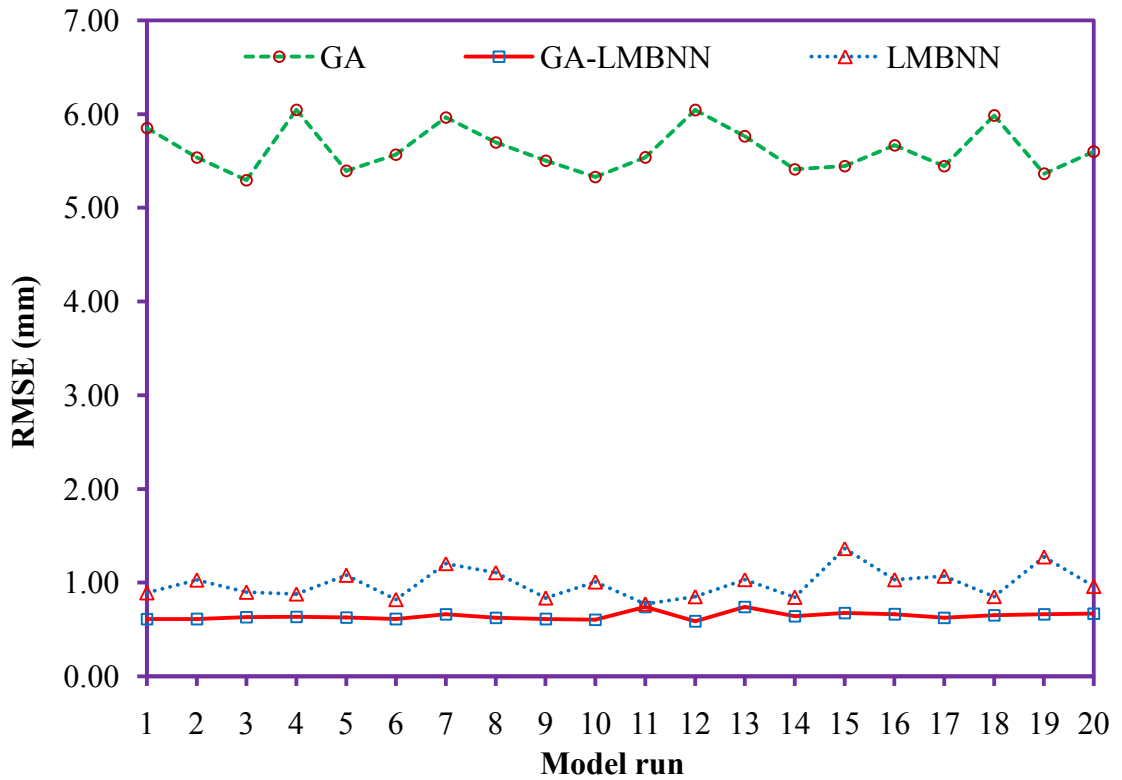
Model	Statistics	Performance metrics						
		RMSE	MAPE	R	E	RSR	PBIAS	
Training	GA	Min.	5.2959	1.9146	0.4340	0.1865	0.7896	-0.3690
		Max.	6.0492	2.3893	0.6141	0.3765	0.9020	0.2726
		Mean	5.6238	2.0973	0.5492	0.2956	0.8385	0.0018
		SD	0.2448	0.1338	0.0527	0.0619	0.0365	0.1803
	GA-LMBNN	Min.	0.5880	0.2496	0.9939	0.9877	0.0784	-0.0061
		Max.	0.7429	0.3311	0.9975	0.9949	0.1108	0.0354
		Mean	0.6461	0.2783	0.9955	0.9910	0.0952	0.0029
		SD	0.0413	0.0233	0.0008	0.0015	0.0078	0.0088
	LMBNN	Min.	0.7726	0.3366	0.9791	0.9586	0.1152	-0.0933
		Max.	1.3650	0.6314	0.9940	0.9880	0.2035	0.0935
		Mean	0.9910	0.4383	0.9891	0.9781	0.1463	0.0026
		SD	0.1610	0.0770	0.0039	0.0071	0.0241	0.0401
Validation	GA	Min.	5.0889	1.8497	0.3561	-0.0383	0.8201	-1.1637
		Max.	6.3224	2.7004	0.5787	0.3274	1.0189	0.5929
		Mean	5.5826	2.1805	0.4844	0.1883	0.8997	-0.5247
		SD	0.2976	0.2103	0.0653	0.0882	0.0480	0.3299
	GA-LMBNN	Min.	0.8332	0.3656	0.9774	0.9523	0.1023	-0.1522
		Max.	1.3177	0.5702	0.9948	0.9895	0.2124	0.1976
		Mean	1.2085	0.5062	0.9816	0.9617	0.1930	0.0158
		SD	0.1229	0.0508	0.0040	0.0086	0.0255	0.0838
	LMBNN	Min.	1.3538	0.5712	0.8135	0.6471	0.1678	-0.4735
		Max.	3.6859	1.1445	0.9865	0.9718	0.5940	0.0997
		Mean	1.9844	0.7617	0.9453	0.8905	0.3173	-0.1168
		SD	0.5771	0.1416	0.0389	0.0736	0.0965	0.1401



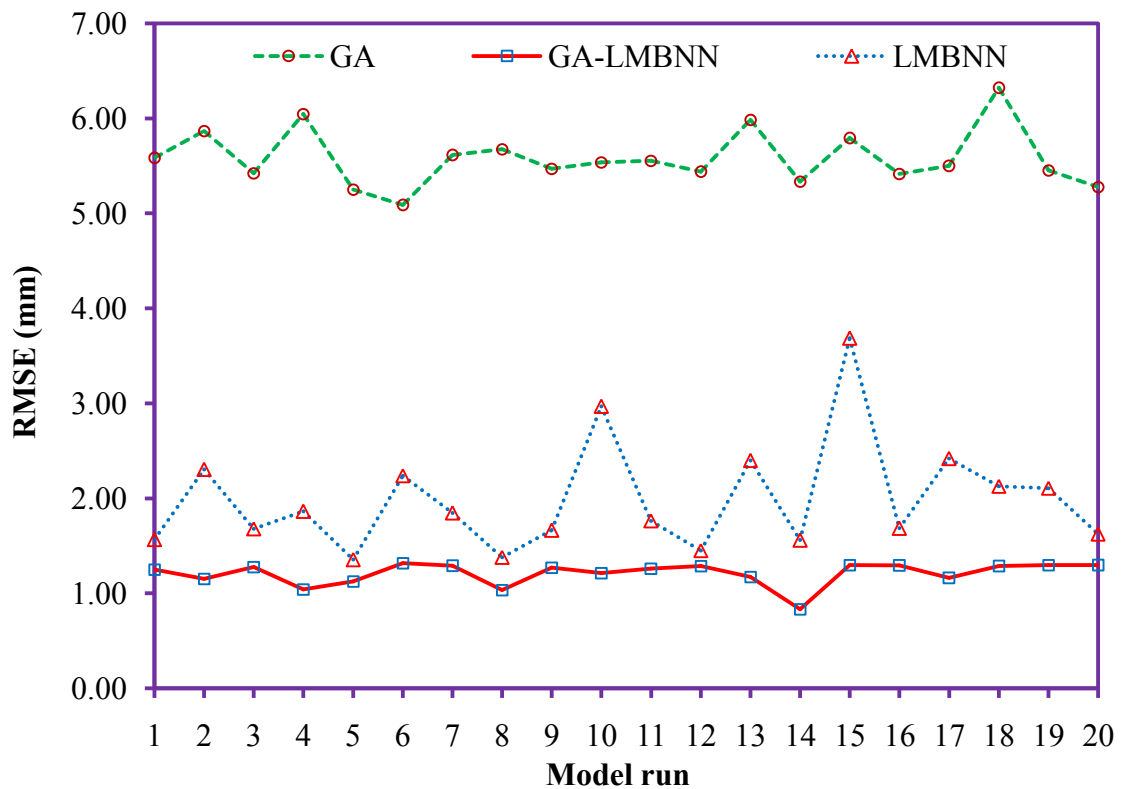
A study of statistics of performance metrics reveals that, the GA model provided higher values of RMSE and MAPE, lower values of R and unacceptable values of RSR and E during training and validation indicating that, the GA model exhibited a relatively poor learning and generalization accuracy in comparison to the GA-LMBNN and LMBNN models. The GA cannot be therefore, regarded as an efficient alternative to the backpropagation algorithms for training the MFNN. In contrast, the GA-LMBNN was shown to provide both learning and generalization accuracy. The conventional LMBNN model on the other hand provided intermediate learning and generalization performance.

The GA model was shown to exhibit the maximum standard deviation for the performance metrics. An intermediate value of standard deviation was obtained for the LMBNN model. It is however, seen that the standard deviation of training and validation performance metrics RMSE, MAPE, R, E and RSR for the GA-LMBNN model are significantly lower than the LMBNN and GA model. Owing to high standard deviation in the performance metrics for GA and LMBNN, a number of fluctuations are seen in the RMSE, MAPE, R, E and RSR performance metric plots (**Figure 5.11** to **Figure 5.20**), indicating a lack of consistency in the learning and generalization performance of these models. Although, the LMBNN and GA-LMBNN models provide comparable prediction accuracy during training yet, a lack of consistency in the prediction accuracy of the LMBNN model is seen, which is shown to be more pronounced during its generalization stage when presented with the validation dataset. The hybridization of GA with LMBNN during training phase is shown to stabilize the performance of the neural network model, augmenting and providing consistency to its learning and generalization accuracy.

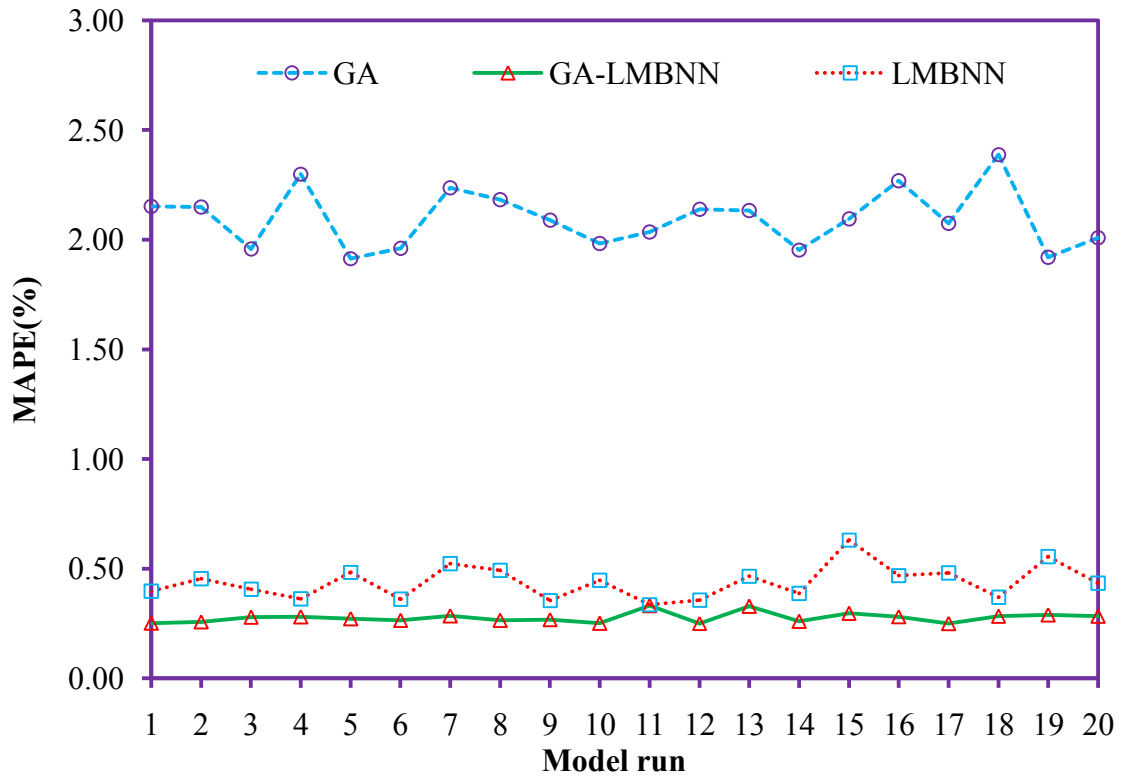
The performance metric PBIAS plot for GA-LMBNN model (**Figure 5.22**) show that, for a particular model run, the PBIAS values follow the same trend i.e., either over-predicting (denoted by positive values) or under-predicting (denoted by negative values) the slump values during training and validation of the model. On the other hand, it is sometimes noticed that PBIAS values for a particular model run of the GA (**Figure 5.21**) and LMBNN model (**Figure 2.23**) are negative during training and positive during validation and vice-versa, indicating their lack of prediction consistency. In nutshell based on the results, the hybrid GA-LMBNN is shown to outperform the GA and conventional LMBNN model.



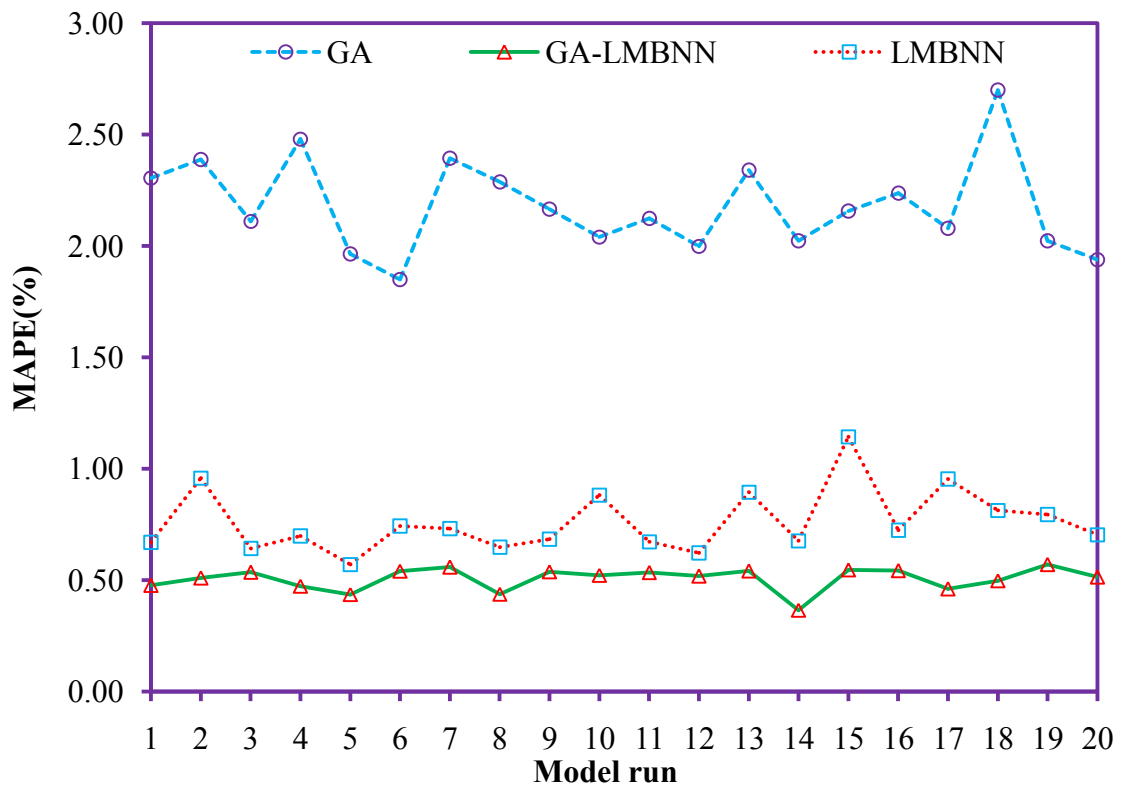
**Figure 5.11:** Plot of training performance metric RMSE for each model run



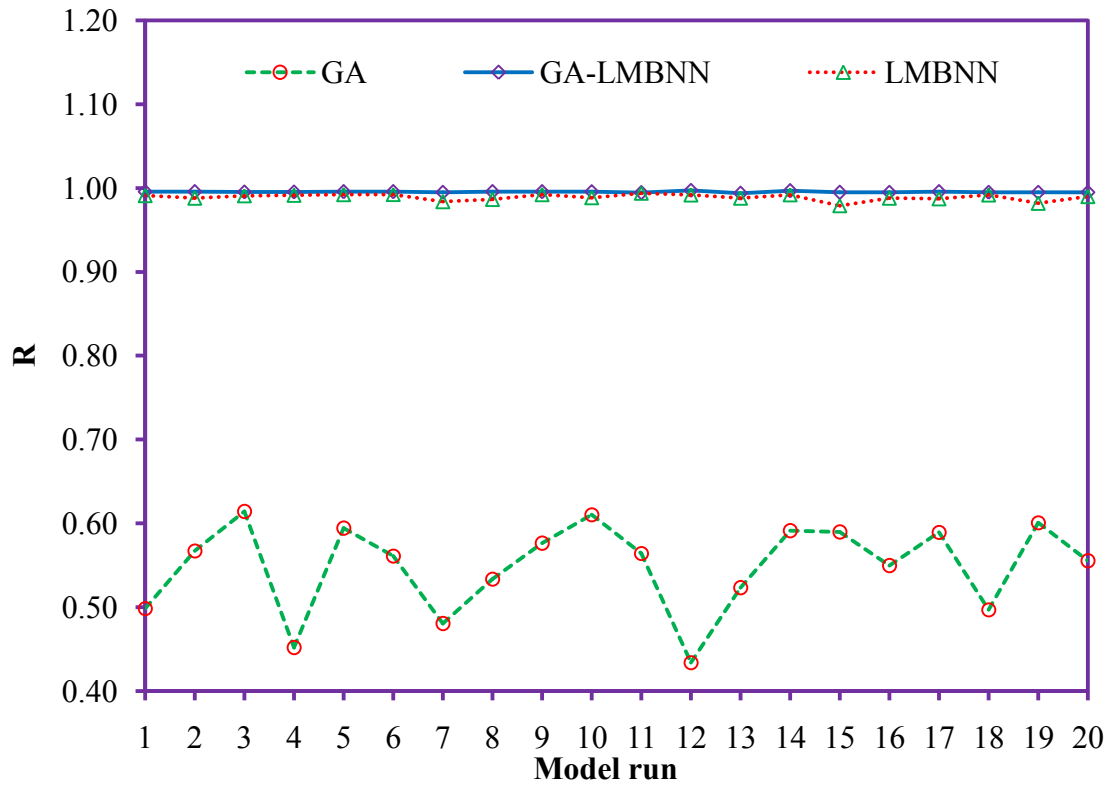
**Figure 5.12:** Plot of validation performance metric RMSE for each model run



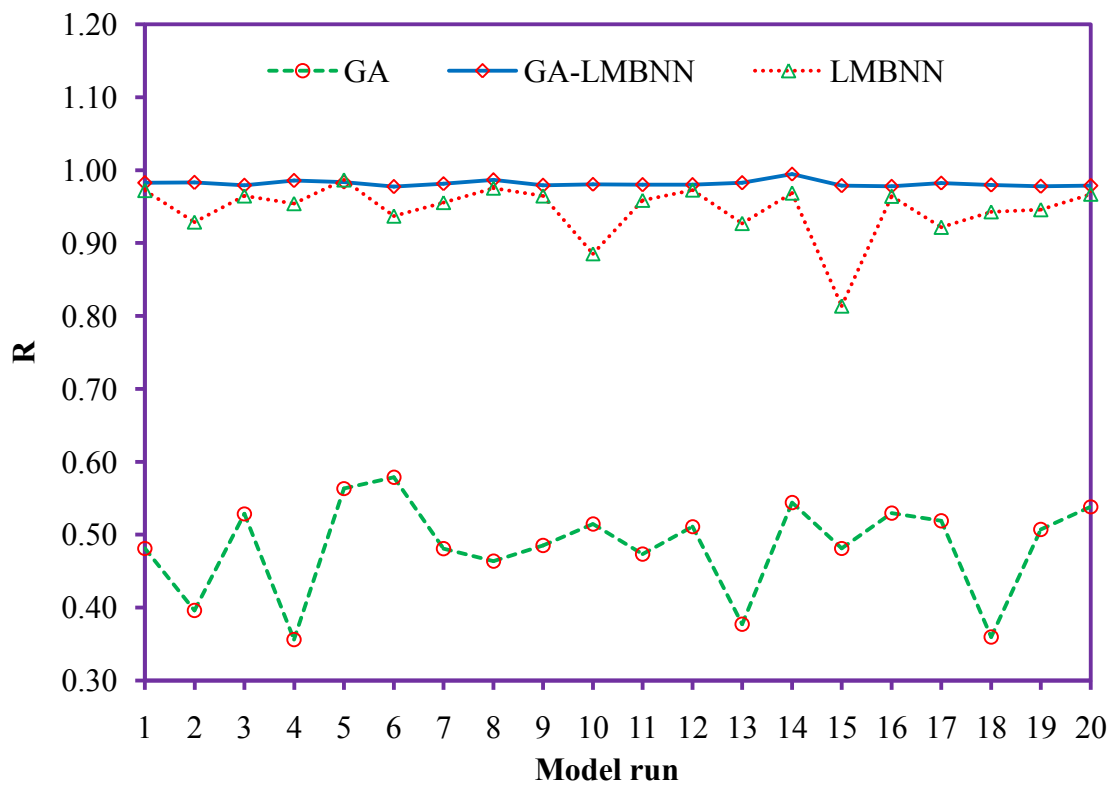
**Figure 5.13:** Plot of training performance metric MAPE for each model run



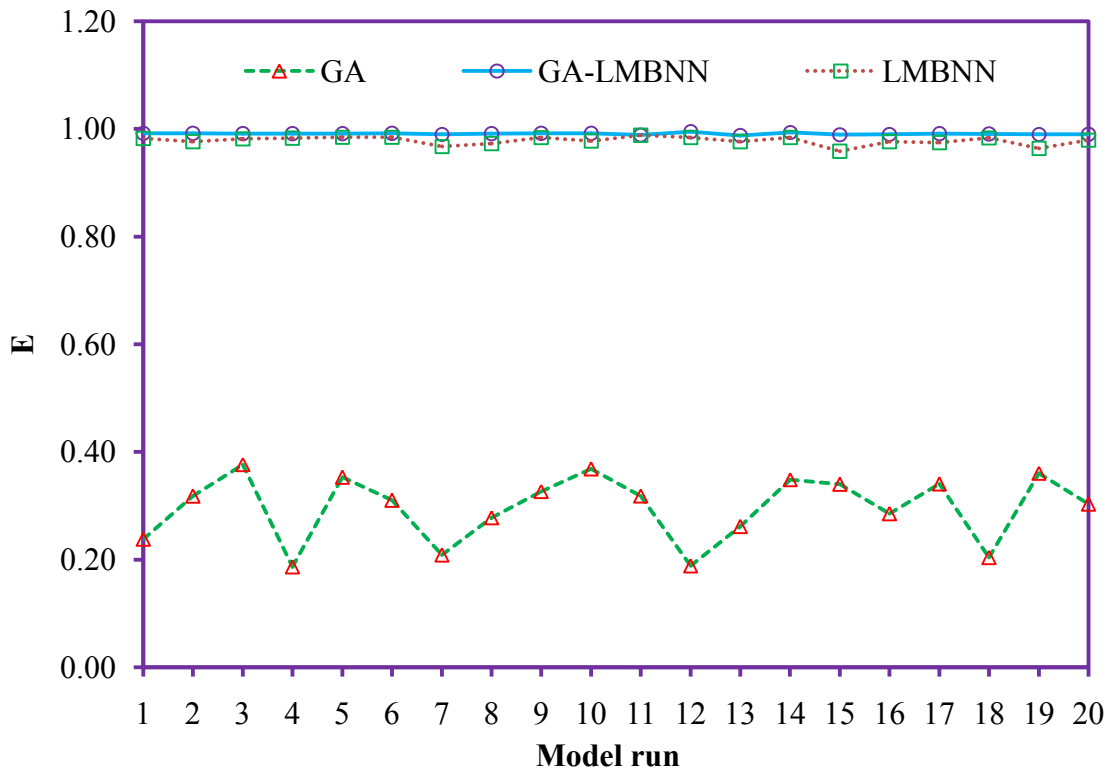
**Figure 5.14:** Plot of validation performance metric MAPE for each model run



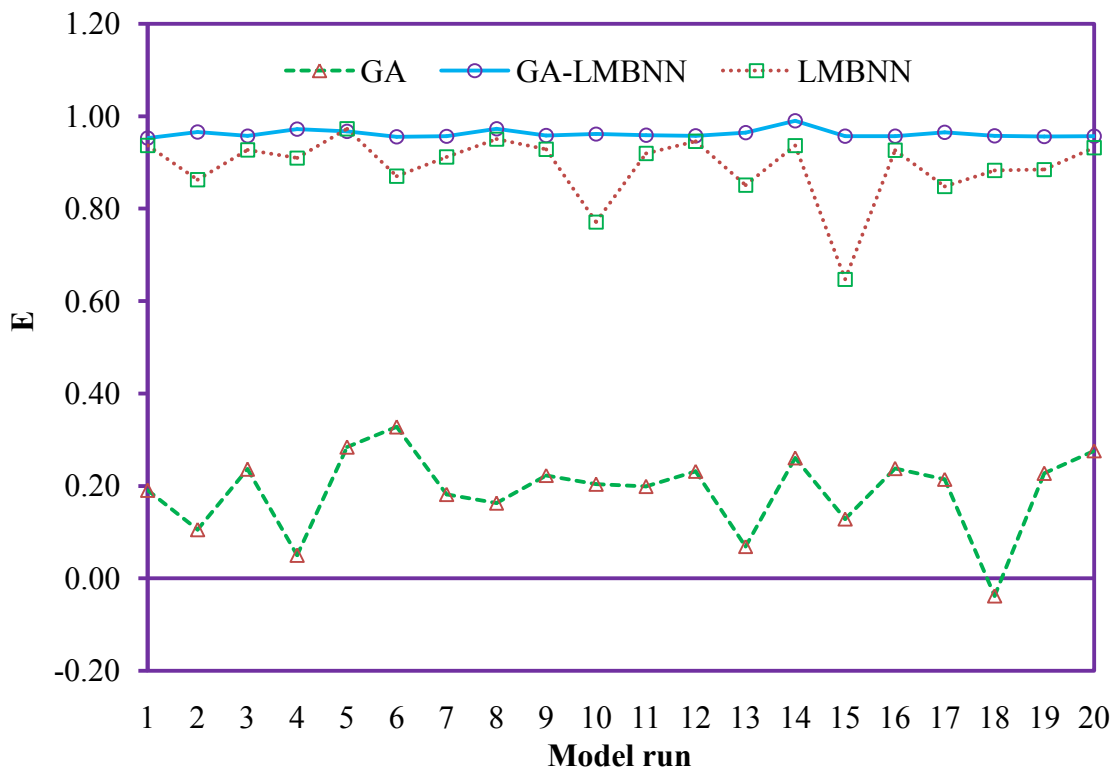
**Figure 5.15:** Plot of training performance metric R for each model run



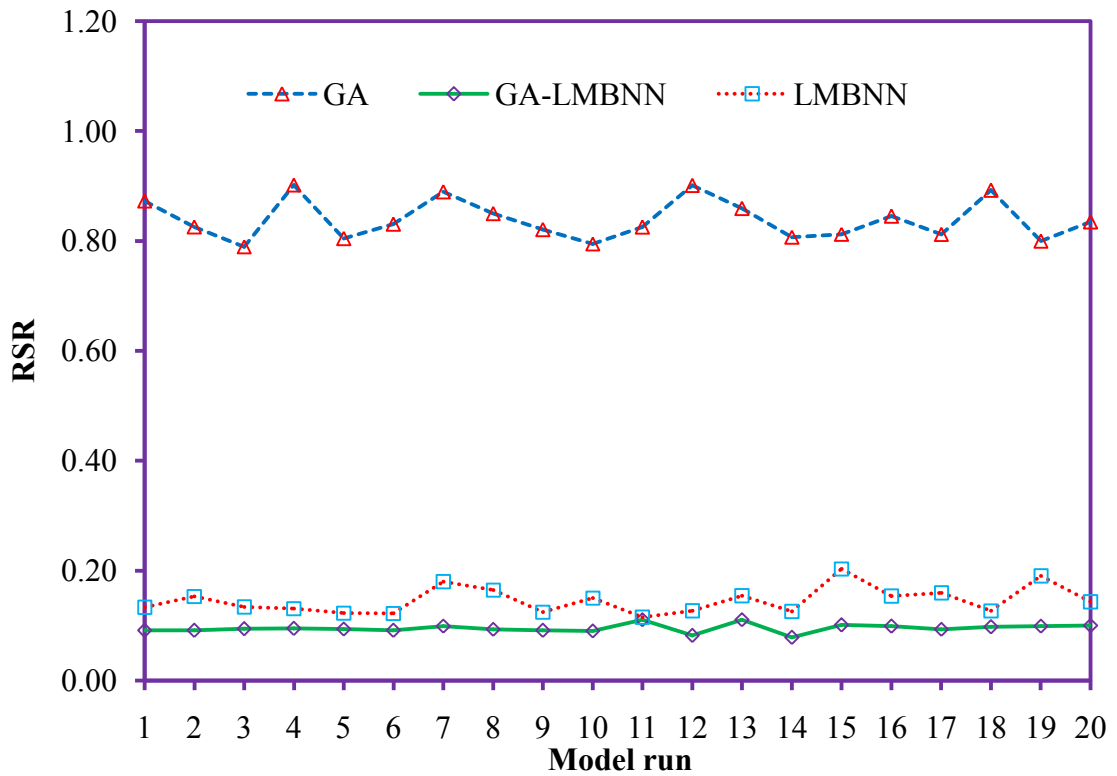
**Figure 5.16:** Plot of validation performance metric R for each model run



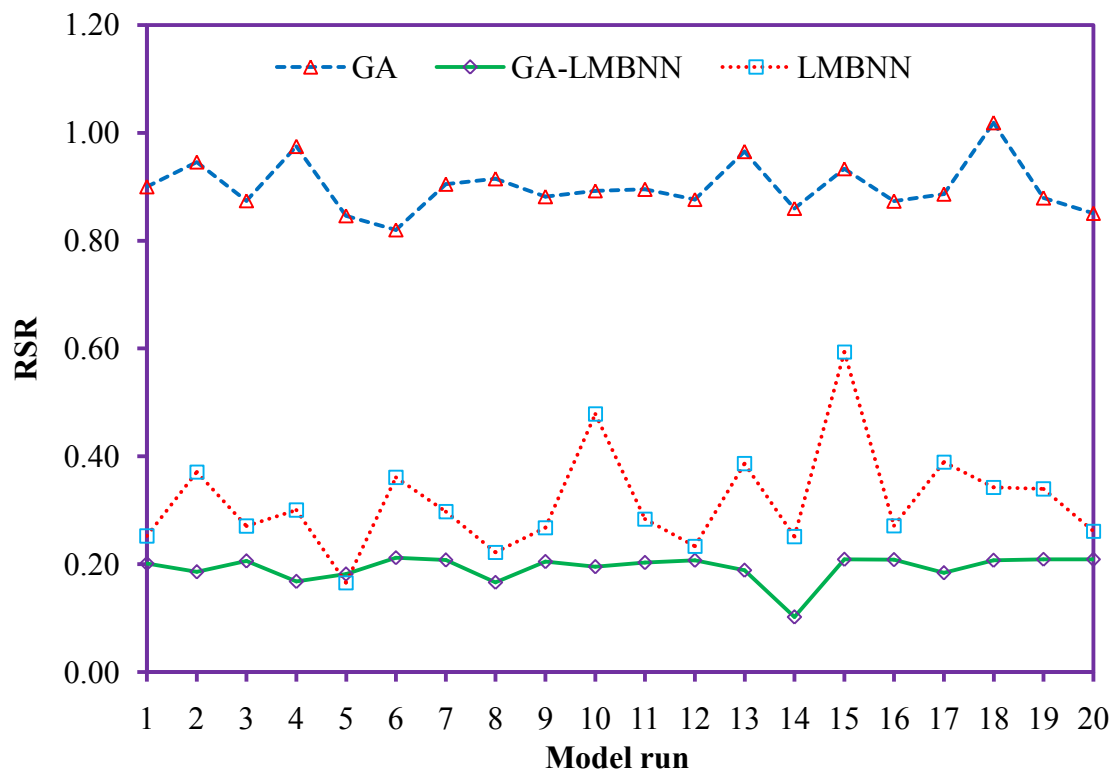
**Figure 5.17:** Plot of training performance metric E for each model run



**Figure 5.18:** Plot of validation performance metric E for each model run



**Figure 5.19:** Plot of training performance metric RSR for each model run



**Figure 5.20:** Plot of validation performance metric RSR for each model run

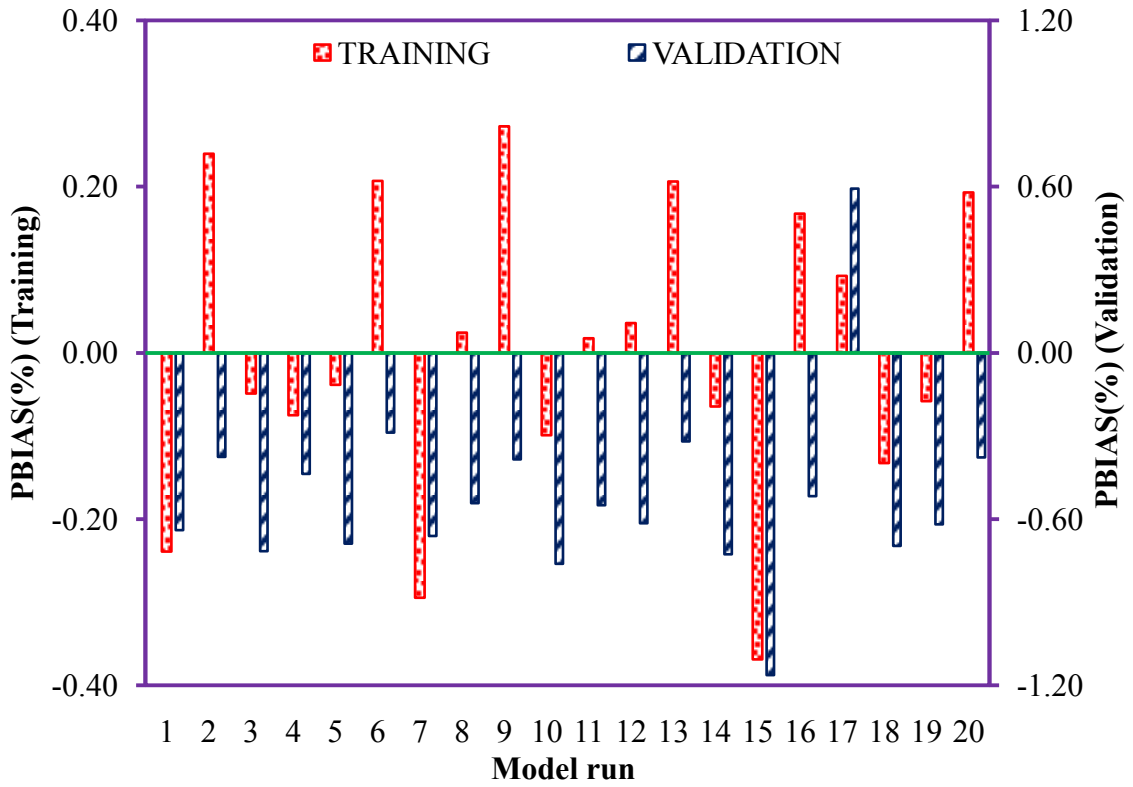


Figure 5.21: Plot of performance metric PBIAS for each run of GA model

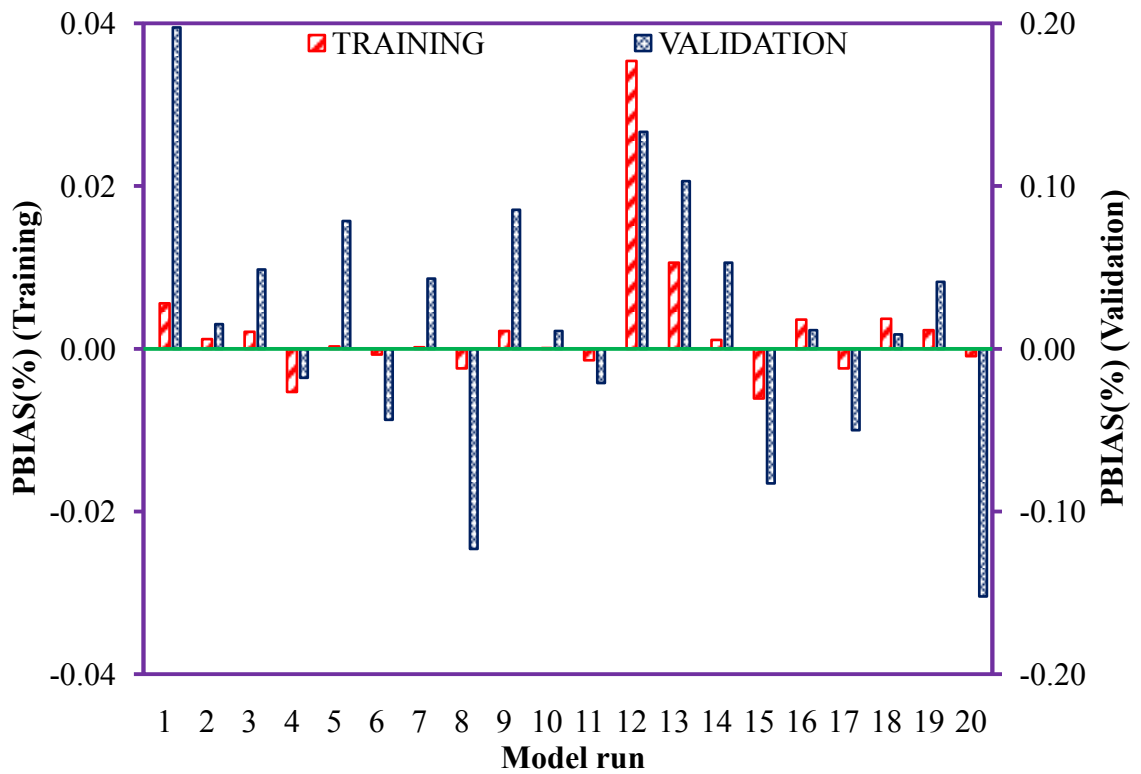
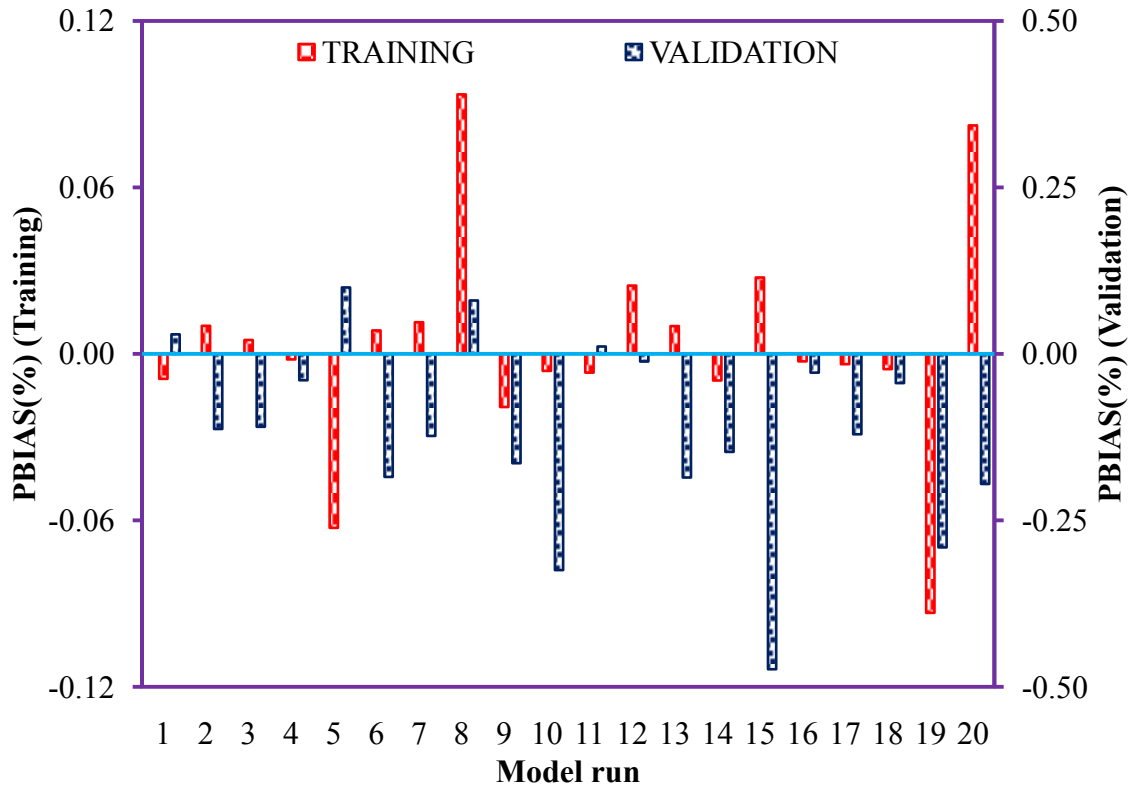


Figure 5.22: Plot of performance metric PBIAS for each run of GA-LMBNN model



**Figure 5.23:** Plot of performance metric PBIAS for each run of LMBNN model

Out of the twenty model runs, the best GA-LMBNN and LMBNN model having the lowest validation RMSE, MAPE, R, E and RSR is selected and further tested using the test dataset. The training, validation and testing performance metrics for the best GA-LMBNN and LMBNN model are shown in **Table 5.7**.

**Table 5.7:** Performance metrics for the best GA-LMBNN and LMBNN models

Model	RMSE (mm)	MAPE (%)	R	E	RSR	PBIAS (%)
<i>Training</i>						
GA-LMBNN	0.6437	0.2604	0.9969	0.9939	0.0784	0.0108
LMBNN	1.0791	0.4841	0.9918	0.9827	0.1314	-0.0627
<i>Validation</i>						
GA-LMBNN	0.8332	0.3656	0.9948	0.9895	0.1023	0.0279
LMBNN	1.3538	0.5712	0.9865	0.9718	0.1678	0.0977
<i>Testing</i>						
GA-LMBNN	1.2926	0.5560	0.9884	0.9761	0.1546	0.1172
LMBNN	3.4652	1.3693	0.9135	0.8283	0.4143	-0.2205



The training, validation and testing performance metrics for the best GA-LMBNN and LMBNN model reveal that, in comparison to the LMBNN model, the hybrid GA-LMBNN model is shown to provide lower RMSE, MAPE and RSR values and higher R and E values, indicating close agreement between the actual and predicted slump values. The PBIAS statistic for LMBNN model gave negative values during training and testing and positive value during validation showing under prediction and over prediction of LMBNN model respectively, indicating model's prediction inconsistency. On the other hand, the GA-LMBNN model is shown to provide lower and consistent PBIAS values during training, validation, and testing of the model.

#### 5.4.2 Regression models

The first order and second regression models for slump were developed using the training dataset and subsequently validated and tested using the validation and test datasets respectively.

##### 5.4.2.1 Training, validation, and testing performance

The training, validation and testing performance of the first order and second order regression model are shown in **Table 5.8**.

**Table 5.8:** Performance metrics for the first order and second order regression models

Model	RMSE (mm)	MAPE (%)	R	E	RSR	PBIAS (%)
<i>Training</i>						
First order	6.4209	2.4709	0.6238	0.3891	0.7816	0.0000
Second order	4.1067	1.6889	0.8661	0.7501	0.4999	0.0000
<i>Validation</i>						
First order	5.4934	2.4630	0.7427	0.5449	0.6746	-0.2868
Second order	4.2243	1.8387	0.8607	0.7309	0.5188	-0.2080
<i>Testing</i>						
First order	5.2948	2.2725	0.7790	0.5992	0.6331	0.0603
Second order	4.4871	2.0026	0.8495	0.7122	0.5365	-0.1353

Compared to the first order regression model, a significant improvement in the performance of the second order regression model is noticeable. This indicates that, as the order of the regression model or its complexity is increased, there is an improvement in the prediction accuracy during training, validation and testing of the model. The PBIAS values show that, the first order regression model under predicts the slump values with the validation data set and over predicts the slump values with the test data set, indicating inconsistent in prediction. An improvement in prediction consistency is noticeable in the case of the second order regression model with lower and negative PBIAS performance metric during the validation and testing of the model.

## 5.5 Neural networks models versus regression models

### 5.5.1 Model prediction error

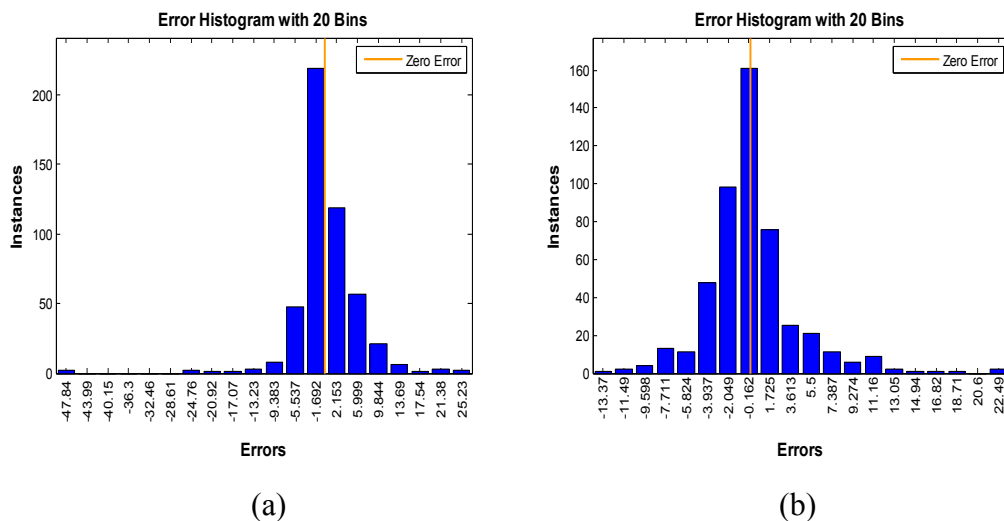
The overall RMC slump data comprising of 493 data pairs were used for computing the prediction error evaluated as the difference between the target slump value and the model's predicted slump value for the slump models namely, first order regression, second order regression, hybrid GA-LMBNN and the conventional LMBNN model. The statistics of the prediction error for the slump models is shown in **Table 5.9**.

**Table 5.9:** Statistics of the prediction error exhibited by the regression and neural network models

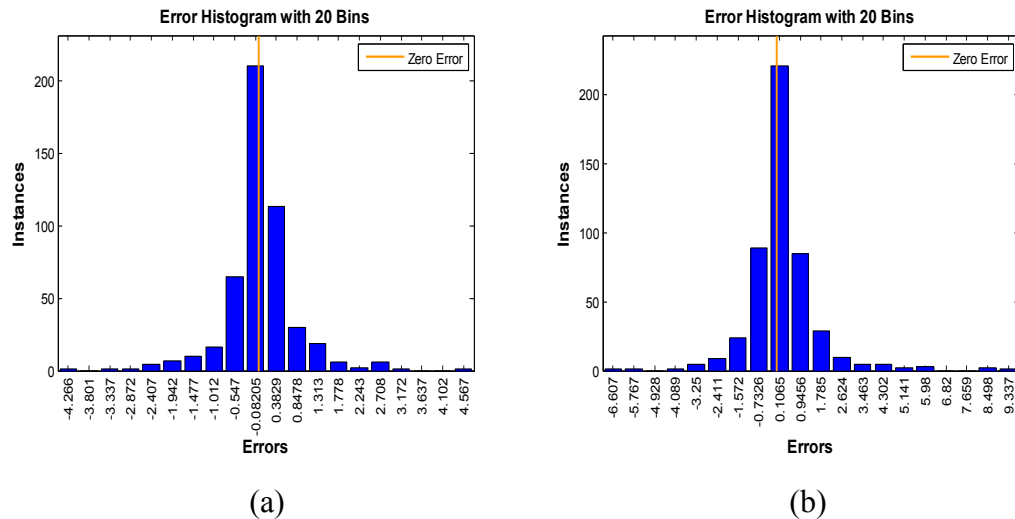
Model	Statistics					
	Min.(mm)	Max.(mm)	Range (mm)	Mean (mm)	SD (mm)	Skew
First order regression	-49.7597	27.1486	76.9083	0.0538	6.1391	-1.8167
Second order regression	-14.3165	23.4289	37.7454	0.0810	4.1863	1.3302
GA-LMBNN	-4.4988	4.7997	9.2985	0.0326	0.8016	0.1552
LMBNN	-7.0262	9.7566	16.7828	0.2474	1.4552	0.7582

The first order and second regression models are shown to exhibit a large range of prediction error. The range of errors is shown to significantly decrease in case of neural network models. The mean value of errors in case of hybrid GA-LMBNN model is found to be very close to zero, indicating a perfect distribution of errors throughout the error range. The standard deviation of model errors show that, in comparison to the regression models, the errors of neural network models are confined in a smaller range, indicating a smaller spread with respect to the mean value and lesser number of outliers.

The error histogram shown in **Figure 5.24(a)** indicate that frequency of errors is maximum in the range [-5.5370 mm, 9.8440 mm] in the case of the first order regression model. The error histogram for second order regression model shown in **Figure 5.24(b)** reveals that most of the errors fall in the range [-3.9370 mm, 3.6130 mm]. An appreciable decrease in error range is noticeable in case of neural network models, with most of the errors falling in the range [-1.0120 mm, 1.3130 mm] in case of GA-LMBNN model (**Figure 5.25(a)**) and [-1.5720 mm, 1.7850 mm] in case of LMBNN model (**Figure 5.25(b)**). The error histograms also show that the maximum frequency of errors close to zero is exhibited by the GA-LMBNN model. The error histograms demonstrate that the neural network models are better modeling tools than the regression models.



**Figure 5.24:** Error histogram for (a) First order regression model; (b) Second order regression model.



**Figure 5.25:** Error histogram for (a) GA-LMBNN model; (b) LMBNN model

With skew value of  $-1.8167$  and  $1.3302$ , the first order and second order regression models exhibit left skewed and right skewed histograms respectively (**Figure 5.24(a)** and **Figure 5.24(b)**). The error histogram for GA-LMBNN model (**Figure 5.25(a)**) is shown to exhibit symmetry of errors with low skew value  $0.1552$ . The error histogram for LMBNN model is shown to be skewed towards right with skew value  $0.7582$  (**Figure 5.25(b)**). The results reveal that the prediction errors in case of hybrid GA-LMBNN model follow nearly a symmetric distribution.

### 5.5.2 Model performance

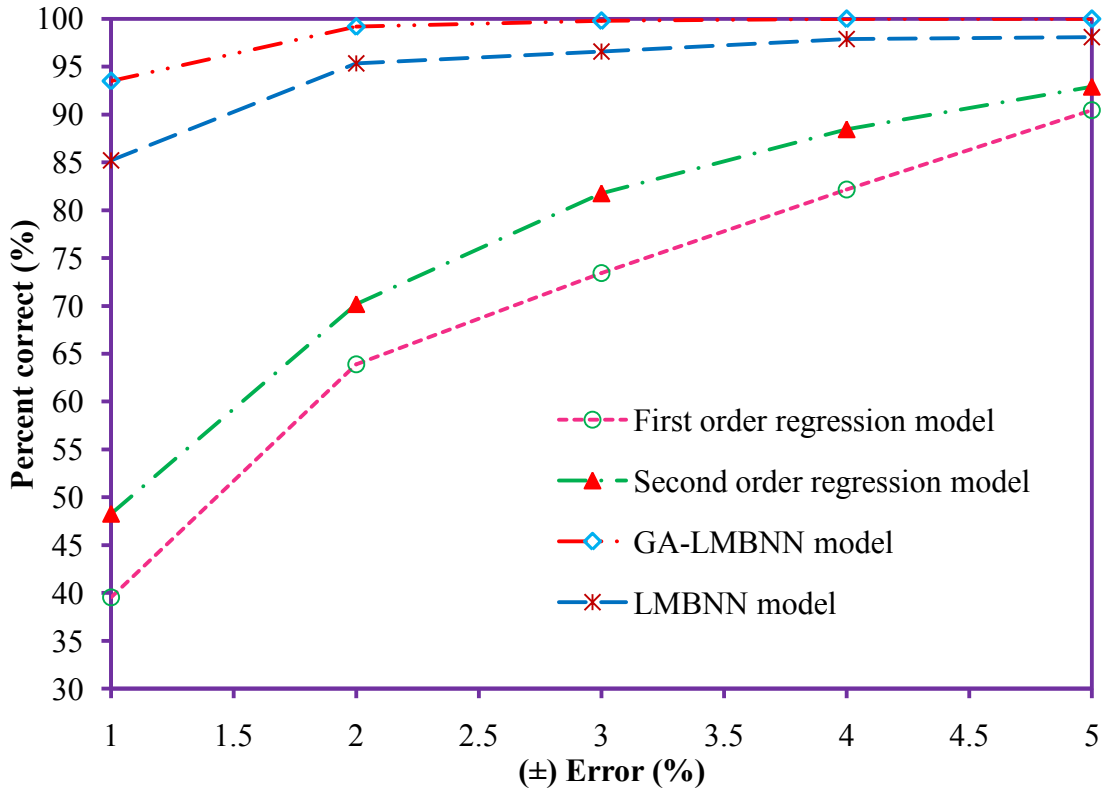
The performance of the regression and neural network models for concrete slump was evaluated using the overall RMC data comprising of 493 data pairs. The performance metrics for the developed slump models are exhibited in **Table 5.10**. Among the concrete slump models, the GA-LMBNN model is shown to exhibit superior accuracy with lower RMSE, MAPE and RSR values along with, higher R and E values. The GA-LMBNN model is also shown to exhibit a very low PBIAS value of  $0.0208\%$  indicating near to optimal prediction accuracy. LMBNN with negative PBIAS value ( $-0.0954\%$ ) is shown to exhibit under-prediction of concrete slump values. Although, regression models are shown to exhibit a lower PBIAS metrics yet, a comparison of all performance metrics taken together demonstrate that the neural network models are far superior to the regression models. The results thus demonstrate the effectiveness and applicability of the artificial neural networks for modeling highly complex and unstructured material behavior of concrete slump.

**Table 5.10:** Model performance for the overall dataset

Model	RMSE (mm)	MAPE (%)	R	E	RSR	PBIAS (%)
<i>Overall</i>						
First order regression	6.2199	2.4456	0.6546	0.4279	0.7564	-0.0881
Second order regression	4.2401	1.7725	0.8572	0.7341	0.5156	-0.0725
GA-LMBNN	0.8015	0.3797	0.9952	0.9905	0.0975	0.0208
LMBNN	1.6942	0.6283	0.9793	0.9576	0.2060	-0.0954

The performance of the developed models for the overall data is also judged by computing the percent correct data falling within a particular threshold error range. In the present study percent, correct data have been evaluated for  $\pm 1\%$ ,  $\pm 2\%$ ,  $\pm 3\%$ ,  $\pm 4\%$  and  $\pm 5\%$  threshold error range. The results of percent correct data are shown in **Figure 5.26**.

The regression models namely, first order and second order are shown to be inefficient at lower threshold error range. The increase in their prediction ability is quite gradual and can correctly predict approximately 90% of the available data within  $\pm 5\%$  threshold error range. The neural network models, on the other hand, are shown to exhibit superior prediction ability at the lower threshold error ranges. LMBNN model is shown to predict correctly nearly 98% of the data within  $\pm 5\%$  error range. However, its percent correct performance is shown to be lower than the hybrid GA-LMBNN model. The GA-LMBNN model is shown to predict correctly 100% of the data within  $\pm 4\%$  error range and is also found to be comparably efficient at smaller error ranges. The percent correct data analysis shows that the neural network models are far efficient than the regression models. Among the developed concrete slump models, the hybrid GA-LMBNN model is shown to possess the best prediction accuracy.



**Figure 5.26:** Percent correct data for concrete slump models

### 5.5.3 Regression plots

The regression equation between independent variable  $x$  (actual slump value) and dependent variable  $y$  (predicted slump value) is expressed as:

$$y = \beta_0 + \beta_1 x \quad (5.1)$$

where,  $\beta_1$  represent the slope and  $\beta_0$  represent the y-intercept of the regression line.

To determine the existence of a significant linear relationship between the actual slump and model predicted slump value, following hypothesis for slope  $\beta_1$  of best fit line are tested:

**(a) Slope  $\beta_1$  is equal to zero**

$$H_0 : \beta_1 = 0 \quad (5.2)$$

$$H_a : \beta_1 \neq 0 \quad (5.3)$$

For the hypothesis shown in **Equation 5.2** and **Equation 5.3**, the regression parameter estimates for slope  $\beta_1$ , t-values and p-value at 0.05 level of significance ( $\alpha$ ) for first order regression, second order regression, GA-LMBNN and LMBNN models are shown in **Table 5.11**.

**Table 5.11:** Regression parameter estimates, t-value and p-value for slope

Model	Slope $\beta_1$	Standard error	t-value	p-value
First order regression	0.4171	0.0217	19.1855	1.24e-61
Second order regression	0.7544	0.0204	36.8875	1.2e-143
GA-LMBNN	0.9929	0.0043	226.5160	0
LMBNN	0.9543	0.0089	107.1447	0

The results exhibited above, show that the t-value for all models is greater than the critical value of t (= 1.9648) at degree of freedom (= 491). Moreover, the p-value is less than assumed level of significance  $\alpha$  (= 0.05). Hence, null hypothesis is rejected and it is inferred that, for all models there exists a linear relationship between the actual slump and the model predicted slump value.

**(b) The slope  $\beta_1$  is equal to unity**

$$H_0 : \beta_1 = 1 \quad (5.4)$$

$$H_a : \beta_1 \neq 1 \quad (5.5)$$

For the above hypothesis, the regression parameter estimates for slope  $\beta_1$ , t-values and p-value at 0.05 level of significance ( $\alpha$ ) for first order regression, second order regression, GA-LMBNN and LMBNN models are shown in **Table 5.12**.

**Table 5.12:** Regression parameter estimates, t-value and p-value for slope

Model	Slope $\beta_1$	Standard error	t-value	p-value
First order regression	0.4171	0.0217	26.8096	3.3627e-98
Second order regression	0.7544	0.0204	12.0118	2.5518e-29
GA-LMBNN	0.9929	0.0043	1.6129	0.1074
LMBNN	0.9543	0.0089	5.1349	4.0758e-07

The results exhibited in **Table 5.12**, show that for first order regression, second order regression and LMBNN models, the t-value is greater than the critical value of t (= 1.9648) at degree of freedom (= 491) and the p-value is significantly less than the level of significance  $\alpha$  (= 0.05). However, for GA-LMBNN the t-value is shown to be less than the critical value of t and the p-value value for is shown to be greater than the level of significance  $\alpha$  (= 0.05). Hence, the null hypothesis is rejected in case of first order regression, second order regression and LMBNN models, indicating that the linear relation between actual slump values and the slump values predicted by these models is not significant. The null hypothesis on the other hand, is accepted for the GA-LMBNN model. It is inferred that, the slope of the regression line for GA-LMBNN model does not significantly differ from unity and therefore, a significant linear relation exists between the actual slump and the GA-LMBNN model predicted slump value.

Following hypothesis for intercept  $\beta_0$  is also tested to assess closeness of best-fit regression line and Actual=Predicted line:

(a) **Intercept  $\beta_0$  is equal to zero**

$$H_0 : \beta_0 = 0 \quad (5.6)$$

$$H_a : \beta_0 \neq 0 \quad (5.7)$$

For the above hypothesis, the regression parameter estimates for intercept  $\beta_0$ , t-values and p-value at 0.05 level of significance ( $\alpha$ ) for first order regression, second order regression, GA-LMBNN and LMBNN models are shown in **Table 5.13**.

**Table 5.13:** Regression parameter estimates, t-value and p-value for intercept

Model	Intercept $\beta_0$	Standard error	t-value	p-value
First order regression	91.8387	3.4354	26.7328	7.77e-98
Second order regression	38.6480	3.2314	11.9602	4.13e-29
GA-LMBNN	1.0828	0.6926	1.5633	0.1186
LMBNN	6.9081	1.4073	4.9087	1.25e-06

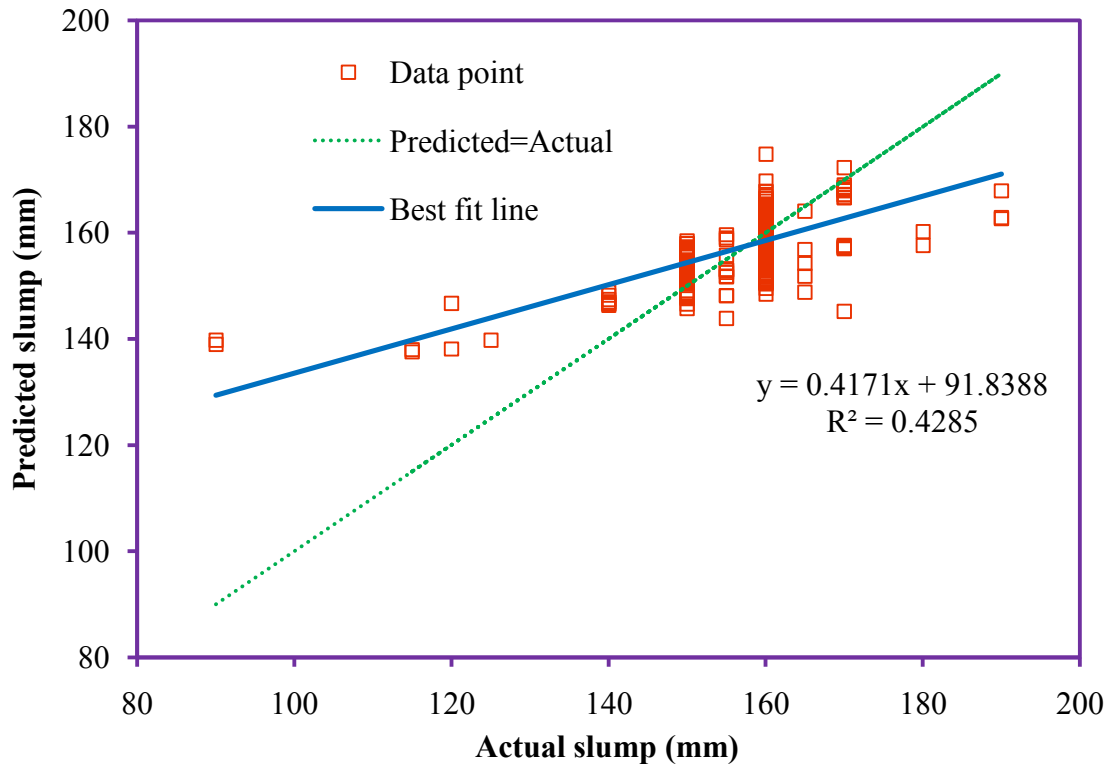
The results exhibited above, show that for first order regression, second order regression and LMBNN models, the t-value is greater than the critical value of



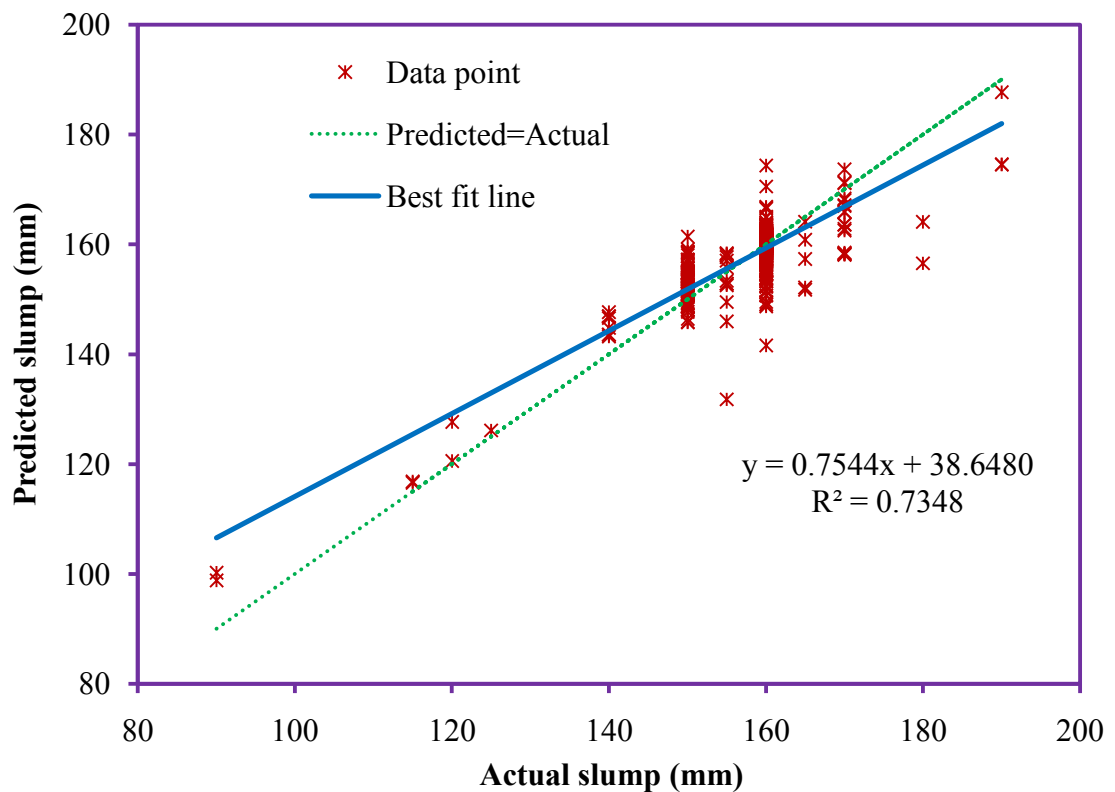
$t$  (= 1.9648) at degree of freedom (= 491) and the  $p$ -value is significantly less than the level of significance  $\alpha$  (=0.05). However, in case of GA-LMBNN model, the  $t$ -value is shown to be less than the critical value of  $t$  and the  $p$ -value is shown to be greater than the level of significance  $\alpha$  (=0.05). Hence, the null hypothesis is rejected in case of first order regression, second order regression and LMBNN models indicating that, the intercept is significant. The null hypothesis on the other hand, is accepted for the GA-LMBNN model, indicating that the intercept of the best-fit regression line for GA-LMBNN model does not significantly differ from zero and therefore, the regression line passes through the vicinity of origin and overlaps the line of equality (Actual=Predicted line).

The results of the hypothesis test were graphically examined by plotting regression plot between the actual and predicted slump values for first order, second order, GA-LMBNN and LMBNN models and are respectively shown in **Figure 5.27** to **Figure 5.30**. The regression plot for first order regression (**Figure 5.27**) shows that the best fit line is significantly away from the line of equality, indicating a poor agreement between the actual and the predicted slump values. The data are shown to be widely scattered with respect to the best-fit regression line. With coefficient of determination  $R^2$  value 0.4285, the regression line represented by the equation  $y = 0.4171x + 91.8388$  is shown to fit only 42.85% data, indicating lack of prediction accuracy of the first order regression model.

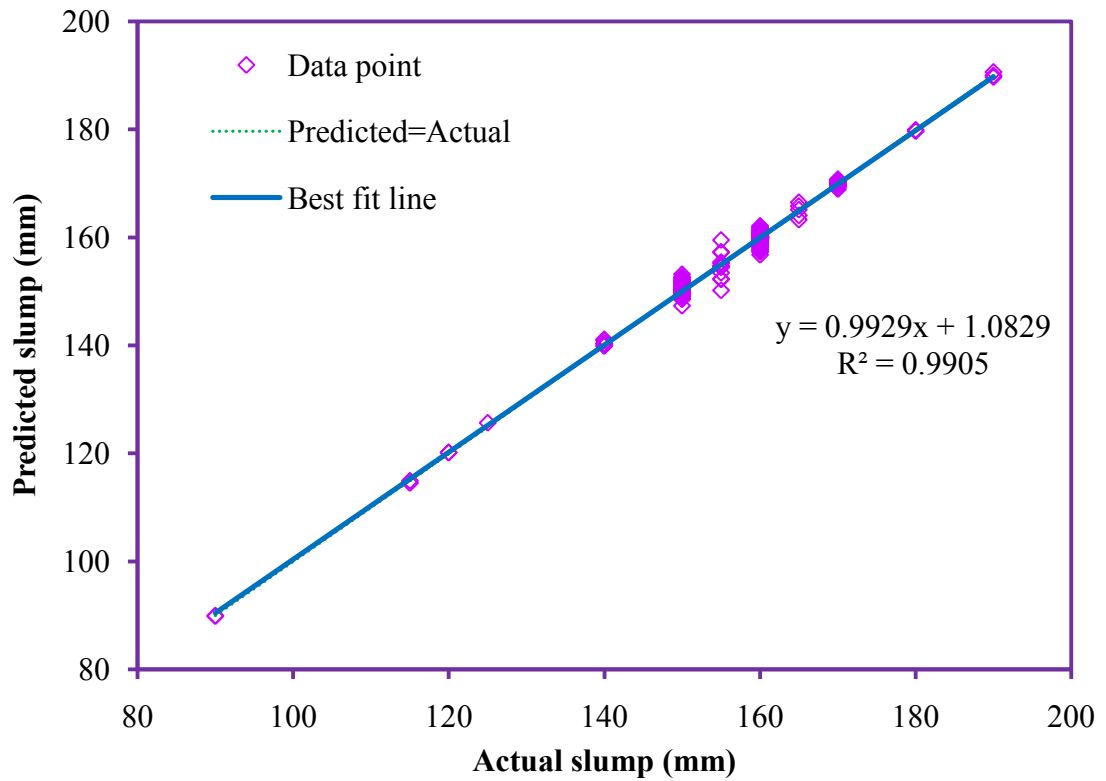
A significant improvement in the regression plot (**Figure 5.28**) is noticeable in the case of the second order regression model. The gap between the best-fit regression line and the line of equality is shown to decrease indicating an improvement in the degree of association between actual and predicted slump values. However, data widely scattered around the best fit line are clearly visible. With  $R^2$  value 0.7348, the regression line represented by the equation  $y = 0.7544x + 38.6480$  is shown to fit 73.48% data. A nearly perfect fit between the actual and predicted slump values is noticeable in the regression plot for hybrid GA-LMBNN model (**Figure 5.29**). The best-fit regression line is shown to overlap the line of equality. The data are shown to be closely spaced with respect to the best-fit regression line. The  $R^2$  value 0.9905 shows a strong agreement between the actual and predicted slump values. With  $R^2$  value 0.9905, regression line represented by the equation  $y = 0.9929x + 1.0829$  is shown to fit 99.05% data.



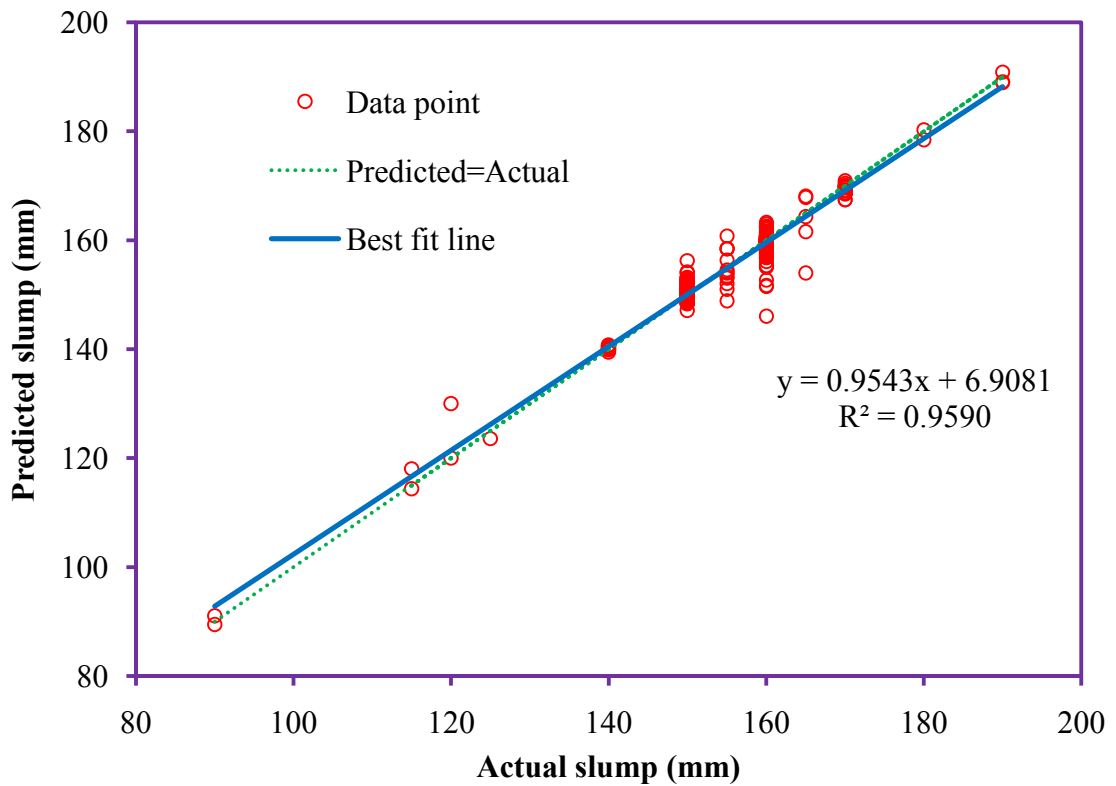
**Figure 5.27:** Regression plot between actual and first order regression model predicted concrete slump



**Figure 5.28:** Regression plot between actual and second order regression model predicted concrete slump



**Figure 5.29:** Regression plot between actual and GA-LMBNN model predicted concrete slump



**Figure 5.30:** Regression plot between actual and LMBNN model predicted concrete slump

In the LMBNN model, a slight difference between the best-fit regression line and the Actual=Predicted line is noticeable (**Figure 5.30**). The scatter of data is significantly lesser than the regression models. With  $R^2$  value 0.9590, the regression equation  $y = 0.9543x + 6.9081$  is shown to fit 95.90% data.

The hypothesis test for the regression plots reveal that, although all models developed for the concrete slump provided a linear relationship between the actual slump and slump values predicted by them but, the hybrid GA-LMBNN model was shown to provide the highest prediction accuracy. The neural network models namely, LMBNN and GA-LMBNN have shown to provide better  $R^2$  values in comparison to regression models, demonstrating their potential to accurately model the complex interactions between the proportions of concrete design mix constituents and the slump value.

## 5.6 Relative importance of RMC design mix constituents on the slump value

The weights of the hybrid GA-LMBNN are harnessed to assess the relative importance of RMC design mix constituents on the slump value. The weights between the input layer and hidden layer and between the hidden layer and output layer were recorded and shown in **Table 5.14** and **Table 5.15** respectively. Using the connection weight approach, the importance and ranking of the concrete design mix constituents, are evaluated and presented in **Table 5.16**.

Based on the values of the relative importance of concrete design mix constituents shown in **Figure 5.31**, following inferences can be drawn:

- a) It can be shown that superplasticizer with an importance of 30.2518%, is the most significant design mix constituent for slump of RMC. The ability of superplasticizer to deflocculate and disperse the cement paste particles, releases the water entrapped by cement particles increasing the water availability to lubricate the mix and as a consequence, the workability of concrete is improved.

The effectiveness of superplasticizers to provide increased initial workability is utilized in the RMC industry to cope with the slump loss during transit of the RMC mix from the manufacturing plant to the construction site, without

having to alter or increase the water-binder ratio prescribed for a customized grade of concrete. Thus, the use of superplasticizers facilitates the transportation of the customized grade of RMC mix for long distances (up to 40-45 km), retaining the workability desired for the construction activity and delaying the setting time of the concrete. Since superplasticizers aid flowability of concrete, they also reduce the effort during pumping of RMC at the construction site.

- b) With relative importance of 21.8040%, pulverized fuel ash (PFA) or fly ash is the second most important design mix parameter influencing the concrete slump. It is shown to exhibit positive influence on the slump value of concrete. This is attributed to the spherical shape of glassy particles present in PFA that act as “ball bearings” to lubricate the concrete mix, improving its rheology whereby, making it more workable.

It can also be shown that, due to higher finer fraction of PFA particles, it exhibits a greater role in workability enhancement as compared to cement. The spherical shape of particles also tends to reduce the friction between concrete mix and pump line, enhancing the pumpability of RMC mix. Apart from this, PFA plays a major role in the RMC industry, as it not only economizes the cost of RMC but also, prolongs the setting time of concrete, thereby reducing the slump loss during the transit from the batching plant to the construction site.

- c) Water and cement content are shown to exhibit almost the same relative importance of 11.6314% and 13.0592% respectively. This is attributed to the fact that, the water-cement ratio is almost fixed for a customized grade of concrete, hence higher the cement content larger will be the water content in the concrete. It is also seen that both water and cement contribute to an increase in the slump of concrete. Water lubricates the concrete mix by forming a thin film around the particles, thereby enhancing its fluidity leading to an increase in workability. Cement paste on the other hand coats the aggregates fills the inter-particle spaces and acts as a lubricating medium to increase workability.

**Table 5.14:** Neural network weights between input layer and hidden layer

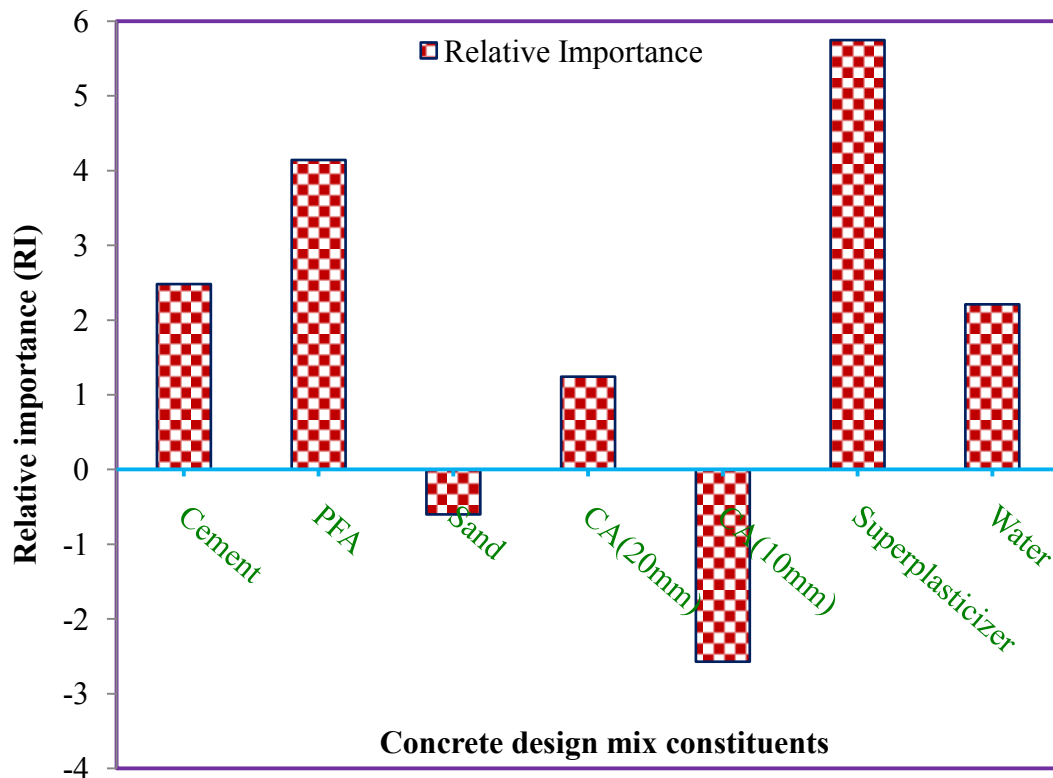
Input	Hidden layer neurons										
	1	2	3	4	5	6	7	8	9	10	11
Cement	-1.6480	1.4052	1.3806	1.0471	-2.1710	-0.5719	1.2100	2.0498	-0.4928	2.4189	1.9326
PFA	0.2520	2.1310	-1.7135	-1.5053	-0.1295	-1.7744	-1.8249	4.7089	0.0934	0.4471	3.0687
Sand	-0.1407	-0.4501	1.2743	0.3785	0.4597	-1.5551	0.8436	-1.8158	-0.5467	-0.8022	-0.2841
CA(20mm)	-0.3761	0.9992	2.9107	0.3402	1.1067	-0.2518	1.2496	-3.1981	-0.6848	-1.1139	-0.6771
CA(10mm)	0.8692	0.2074	1.2252	-0.3416	0.8202	3.5798	0.3149	-0.1433	3.3713	-1.2856	0.7669
Superplasticizer	0.0257	4.1526	0.5368	-0.2901	0.6781	0.5362	-0.0495	0.9163	0.8631	-0.2956	0.3934
Water	-0.1197	1.2880	-0.1716	-0.0094	-0.2881	0.1445	-0.1579	0.9020	-0.2662	0.4341	0.0315

**Table 5.15:** Neural network weights between the hidden layer and output layer

Output	Hidden layer neurons										
	1	2	3	4	5	6	7	8	9	10	11
Slump	1.5761	1.5499	1.0166	4.5679	1.1440	-0.3565	-4.0624	-0.3415	-0.4023	1.2363	0.7348

**Table 5.16:** Relative importance of concrete design mix constituents on concrete slump

Concrete design mix constituents (Inputs)	Relative importance (RI)	RI (%)	Ranking as per RI
Cement	2.4805	13.0592	4
PFA	4.1415	21.8040	2
Sand	-0.6023	3.1709	7
CA(20mm)	1.2411	6.5341	6
CA(10mm)	-2.5734	13.5483	3
Superplasticizer	5.7461	30.2518	1
Water	2.2093	11.6314	5

**Figure 5.31:** Relative importance of concrete design mix constituents on the slump value

- d) Despite quality checks, in actual practice during crushing operation and handling of aggregates, undersized particles in the range of 4mm to 6mm and aggregate dust, gets inadvertently mixed with the 10mm coarse aggregates. In rainy season, the presence of undersized particles increases, as fine silt and murrum particles get stick with smaller particles present in 10mm sized aggregates which, in spite of washing is not completely removed. Moreover, the elongation and flakiness indices of the aggregates increases as the maximum size of aggregate reduces from 20mm to 10mm (**Pandurangan and Kothandaram, 2012**). The observation is supported by the gradation analysis of two samples collected from a RMC plant shown below, that reveal the presence of undersized particles and aggregate dust in 10mm sized aggregates.

**Sample-I**

IS Sieve (mm)	Weight retained (gms)	Retained (%)	Cumulative retained (%)	Cumulative passing (%)	As per IS:383 limits	Remarks
12.50	0	0	0	100	100	Presence
10.00	100	3.28	3.28	96.72	85-100	of
<b>4.75</b>	<b>1840</b>	<b>60.37</b>	<b>63.65</b>	<b>36.35</b>	<b>0-20</b>	undersized
2.36	758	24.87	88.52	11.48	0-5	particles
pan	350	11.48	100.00	0		and aggregate dust

**Sample-II**

IS Sieve (mm)	Weight retained (gms)	Retained (%)	Cumulative retained (%)	Cumulative passing (%)	As per IS:383 limits	Remarks
12.50	0	0	0	100	100	Presence
10.00	50	1.67	1.67	98.33	85-100	of
<b>4.75</b>	<b>2148</b>	<b>71.72</b>	<b>73.39</b>	<b>26.61</b>	<b>0-20</b>	undersized
2.36	597	19.93	93.32	6.68	0-5	particles
pan	200	6.68	100.00	0		and aggregate dust



It is known that, the coarse aggregates having large size possess smaller surface area, demanding lesser paste for surface coating and vice-versa. Presences of flaky and elongated particles also increase the surface area of aggregates. Moreover, on account of the undersized particles, aggregate dust and silt fines present in 10mm sized aggregates, the voids left by 20mm sized aggregates are not completely filled and are therefore required to be filled using the cement paste to produce a cohesive mix. Therefore, in comparison to 10mm sized aggregate more cement paste is available for lubrication in case of 20mm sized aggregates.

In view of the above, CA(20mm) constituent in concrete is shown to increase the concrete slump attributing 6.5341% influence on slump value. Whereas, CA(10mm) owing to the presence of elongated, flaky, undersized particles and particles covered with silt and murrum, is shown to decrease the slump and provide a significant importance of 13.5483% on the slump value.

- e) Sand content is shown to exhibit a slight negative importance of 3.1709%. Due to declining sources of Natural River sand, it is difficult to procure the desired quantity of sand from a single source. Sand sourced from different quarries affects the particle grading. The findings are supported by the gradation analysis of fine aggregate samples obtained from the RMC industry shown below, that reveal the change in particle grading due to presence of either higher percentage of fine particles or coarse particles.

### Sample-I

IS (mm)	Seive	Weight retained (gms)	Retained (%)	Cumulative retained (%)	Cumulative passing (%)	As per IS:383 limits	Remarks
10.000		0.00	0.00	0.00	100.00	100	The
4.750		30.00	3.01	3.01	96.99	90-100	percentage
2.360		48.00	4.81	7.82	92.18	75-100	of fine
1.180		134.00	13.44	21.26	78.74	55-90	particles is
<b>0.600</b>		<b>95.00</b>	<b>9.53</b>	<b>30.79</b>	<b>69.21</b>	<b>35-59</b>	greater than
0.300		400.00	40.12	70.91	29.09	08-30	the
0.150		200.00	20.06	90.97	9.03	0-10	prescribed
pan		90.00	9.03	100.00	0.00		limit.

**Sample-II**

IS (mm)	Seive	Weight retained (gms)	Retained (%)	Cumulative retained (%)	Cumulative passing (%)	As per IS:383 limits	Remarks
10.000		12.00	1.23	1.23	98.77	100	The
4.750		123.00	12.58	13.80	86.20	90-100	percentage
2.360		108.00	11.04	24.85	75.15	75-100	of coarse
1.180		342.00	34.97	59.82	40.18	55-90	particles is
<b>0.600</b>		<b>102.00</b>	<b>10.43</b>	<b>70.25</b>	<b>29.75</b>	<b>35-59</b>	greater than
0.300		201.00	20.55	90.80	9.20	08-30	the
0.150		76.00	7.77	98.57	1.43	0-10	prescribed
pan		14.00	1.43	100.00	0.00		limit.

Very fine sand, because of their larger surface area demand more cement paste for lubrication thereby, requiring more water to produce a concrete of given workability. High silt content, very rarely found in sands also increases the water demand for a given workability. Very coarse sand particles on the other hand, however, decrease the surface area but, increase particle interference with coarse aggregates leading to decrease in the workability of the concrete mix. Since fine aggregates offer a greater contribution to the surface area, it is shown that the workability of concrete is sensitive to the grading of the fine aggregates.

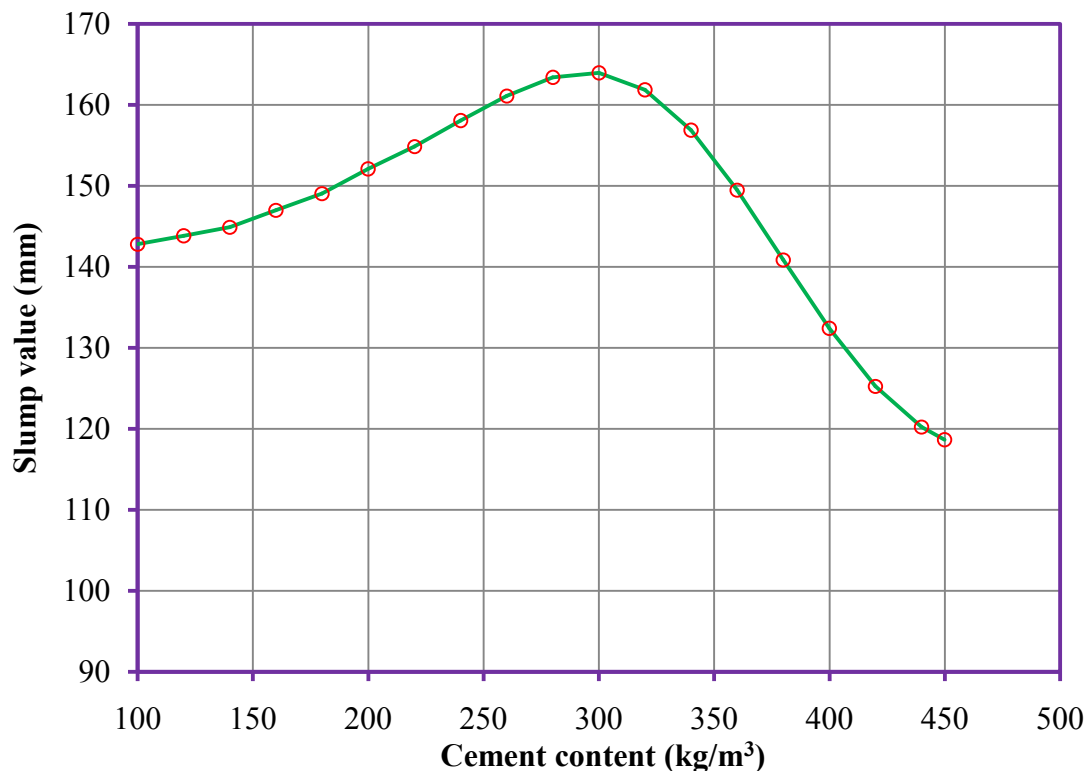
### 5.7 Response trace plots

The response trace plots for the design mix constituents of RMC namely, cement, PFA, sand, CA (20 mm), CA (10 mm), superplasticizer and water, were plotted by varying the proportion of each constituent and predicting the slump value using the trained hybrid GA-LMBNN model. The response trace exhibit the effect of a change in the content of each concrete design mix constituent on the slump such that the total volume of the mix (1 m<sup>3</sup>) remains unchanged. Therefore, a decrease or increase in the content of any concrete design mix constituent with respect to the reference design mix proportion would proportionately either increase or decrease the volume of other constituents.

The response trace plots exhibited in **Figure 5.32** to **Figure 5.38** reveal that:

- a) The slump value is shown to increase gradually with cement content and beyond cement content  $300 \text{ kg/m}^3$  it starts to decrease sharply (**Figure 5.32**). At low cement content, the volume of cement paste is not adequate to completely fill, lubricate and coat the aggregate particles, resulting in lower workability. An increase in the cement content provides more lubrication to the concrete mix, enhancing its slump value. As cement content is increased, the water to binder ratio decreases. However, the water demand to aid hydration of cement increases.

A certain limit of cement content ( $300 \text{ kg/m}^3$  in the present study) is reached wherein the water required for cement hydration exceeds the water available to aid the consistency of the concrete mix. Hence, as the cement content is increased beyond  $300 \text{ kg/m}^3$ , the resulting mix exhibits cohesiveness but lacks the ability to flow, leading to a decrease in the slump value.

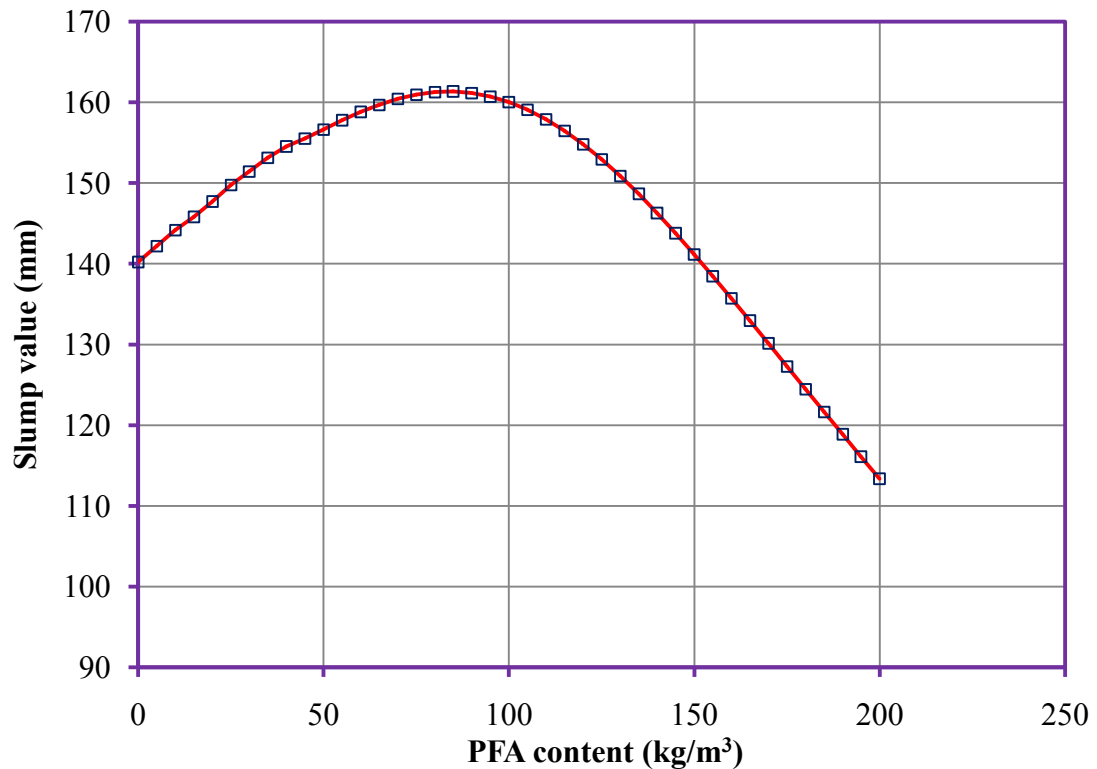


**Figure 5.32:** Response trace of slump value plotted against cement content

- b) A sharp increase in the slump value with an increase in PFA content is noticeable (**Figure 5.33**). The rising limb of the response trace plot for PFA exhibits a sharper increase in slump values in comparison to cement. This is

attributed to higher finer fraction of PFA particles having spherical shape and glassy surface that lead to greater workability enhancement. The smaller particles of PFA also help in reducing segregation, improving the cohesiveness of the mix. However, if the PFA content is increased beyond  $90 \text{ kg/m}^3$ , a sharp decrease in the slump value is noticed.

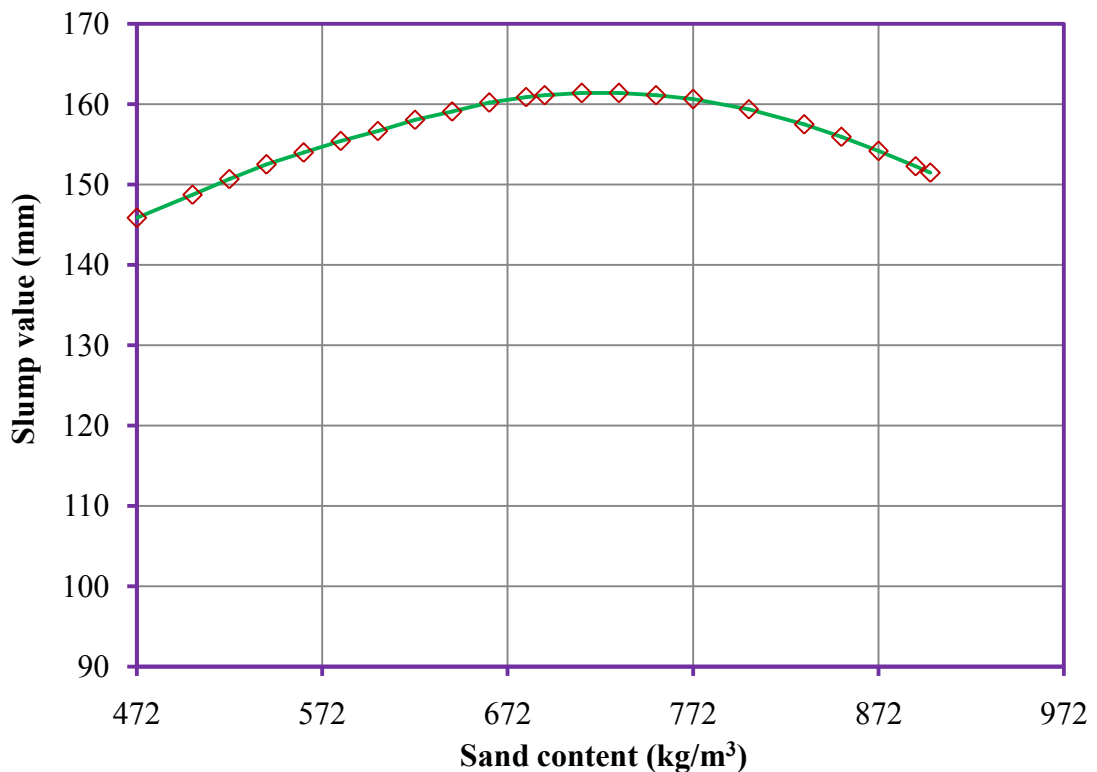
As PFA content is increased, the water to binder ratio decreases. A stage is reached wherein water content becomes inadequate to maintain the consistency of the concrete mix. Hence, any addition in PFA content beyond  $90 \text{ kg/m}^3$  in the present study increase the cohesiveness of the concrete mix at the expense of reduction in consistency making the concrete mix sticky and difficult to finish.



**Figure 5.33:** Response trace of slump value plotted against PFA content

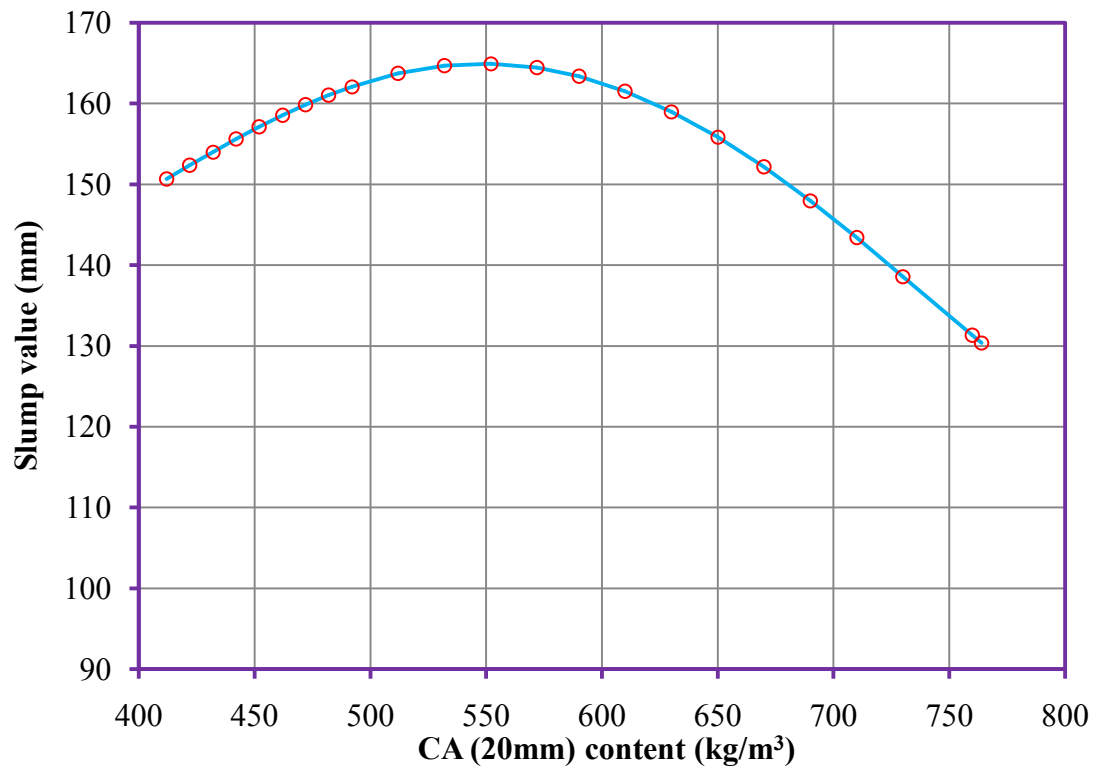
- c) The slump value increases and then decreases very gradually with increase in sand content (**Figure 5.34**). The flatness of the plot is attributed to the sand procured from different natural sources that cause deviation in the uniformity of its particle grading, which is supported by particle gradation shown in **Section 5.6**. As a consequence sometimes finer particles and sometimes coarser particles predominate the particle grading causing either increase in

the water demand or particle inference leading to decrease in workability. However, at lower sand content, the smooth and rounded particles present in Natural River sand tend to increase the workability of concrete. An increase in fine aggregate content beyond  $732 \text{ kg/m}^3$  increases the fine to coarse aggregate ratio thereby, appreciably increasing the demand for cement paste for a given consistency. Therefore, an increase in the sand content beyond  $732 \text{ kg/m}^3$  increases cohesiveness but decreases the consistency of the concrete mix.



**Figure 5.34:** Response trace of slump value plotted against sand content

- d) An increase in slump is noticed up to CA(20mm) content  $572 \text{ kg/m}^3$  (**Figure 5.35**), which is attributed to smaller surface area of large aggregate size that require less cement paste for coating their surface whereby, more cement paste is left for consistency improvement. However, beyond CA (20 mm) content  $572 \text{ kg/m}^3$ , the concrete slump begins to decrease. This is attributed to an increase in aggregate content without increasing the cement paste whereby, the available cement paste becomes insufficient to coat completely and lubricate all aggregate particles, resulting in a decrease in slump value.



**Figure 5.35:** Response trace of slump value plotted against CA (20mm) content

- e) The gradation analysis shown and discussed in **Section 5.6**, reveal that the undersized particles, silt fines, and aggregate dust get inadvertently mixed with CA(10mm). With an increase in CA(10mm) content, the proportion of finer particles in the concrete mix increases, requiring more paste for a given consistency. Moreover, the presence of flaky and sharp angular particles with rough surfaces in CA(10mm) aggregate, cause an interlocking effect that result in restraining the flowability of concrete. As a consequence, the slump value is shown to decrease with increase in CA(10mm), which becomes more pronounced as the CA(10mm) content is increased beyond 490 kg/m<sup>3</sup> (**Figure 5.36**).
- f) With an increase in superplasticizer dosage, the slump value rises sharply which is attributed to the deflocculating and dispersing effect of superplasticizer that makes concrete flowable. However, beyond a certain dosage (3.50 kg/m<sup>3</sup> in the present study), the superplasticizer ceases to further improve the slump value (**Figure 5.37**). It is also shown that beyond an optimal dosage, a very slight decrease in the slump value is noticeable which is attributed to bleeding and segregation caused by excessive dosage of

the superplasticizer that affects the cohesiveness as well as the uniformity of the concrete mix.

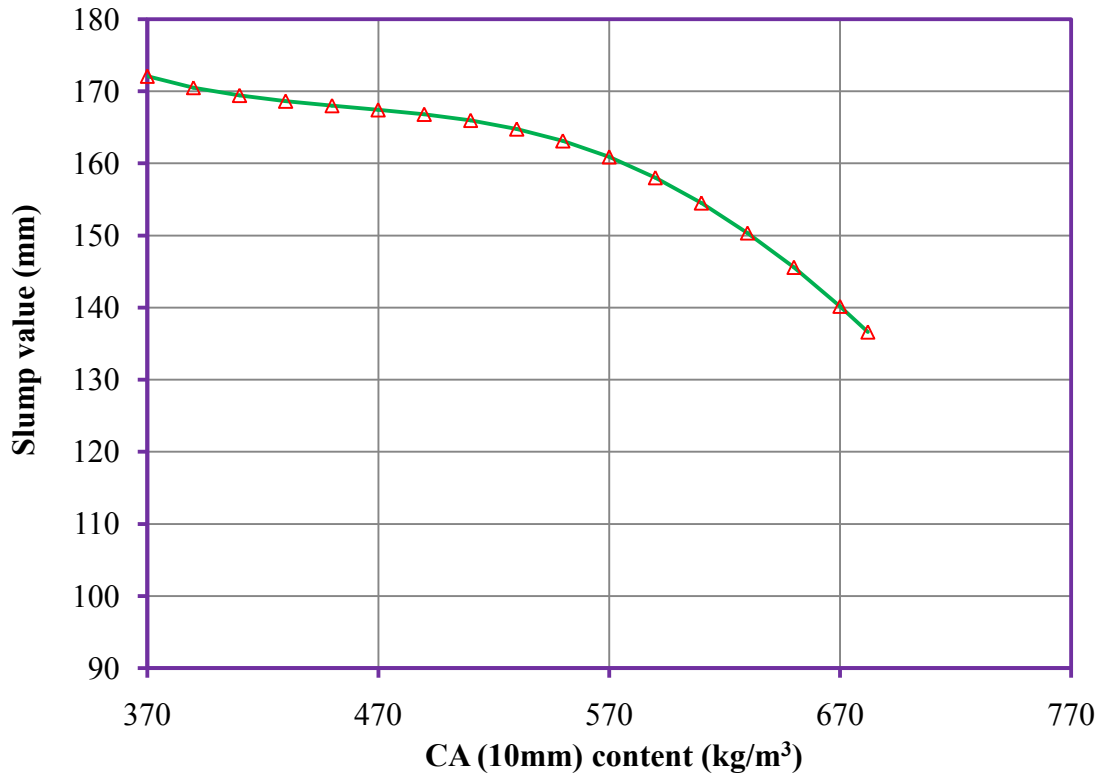


Figure 5.36: Response trace of slump value plotted against CA (10mm) content

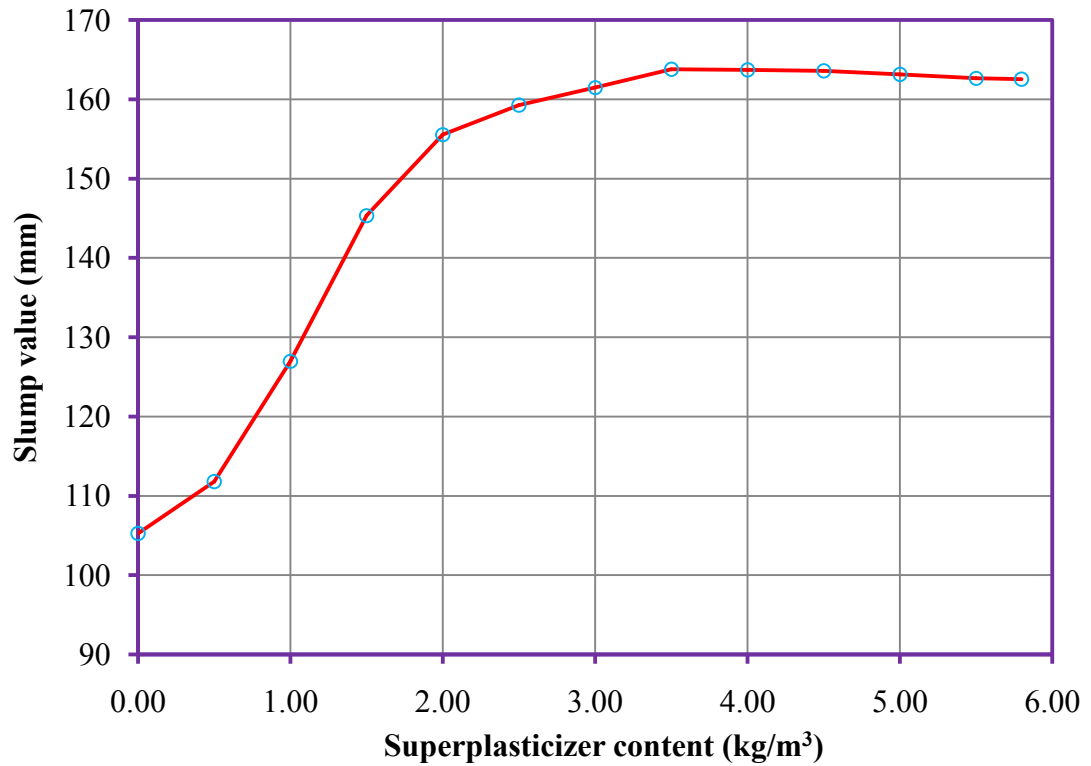
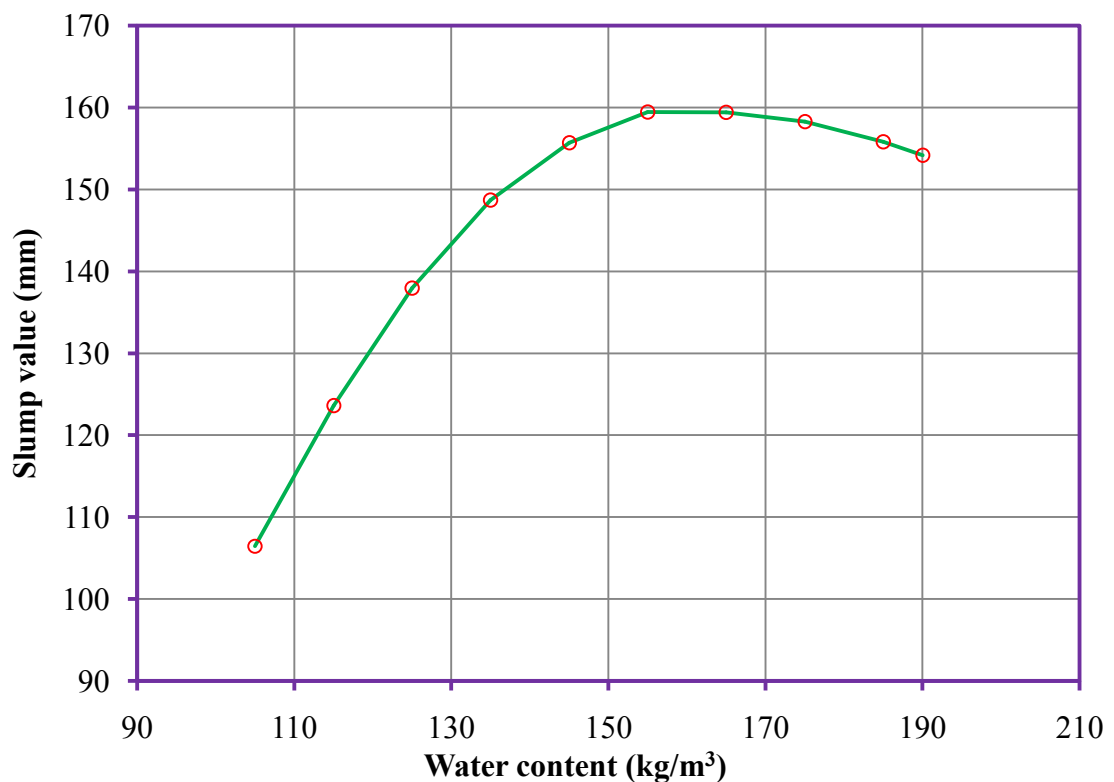


Figure 5.37: Response trace of slump value plotted against superplasticizer content

- g) Increase in water content adds to the fluidity of the concrete mix by lubricating the particles and therefore, as the water content is increased, a sharp rise in the slump value is noticeable (**Figure 5.38**). However, beyond a certain water content ( $155 \text{ kg/m}^3$  in the present study), a slight decrease in slump value is noticeable which is attributed to an increase in fluidity but decrease in the density of concrete, allowing the heavier particles to settle down and lighter particles to move to the surface of concrete mix, leading to non-distribution of particles throughout the concrete cross-section, decreasing the cohesiveness, thereby reducing the slump value.



**Figure 5.38:** Response trace of slump value plotted against water content

### 5.8 Performance and reliability of the concrete slump model for different RMC batching plant

To assess the universal applicability of the best performing hybrid GA-LMBNN model for the concrete slump, it was utilized to predict the slump value for the concrete design mix proportions collected from second RMC batching plant.



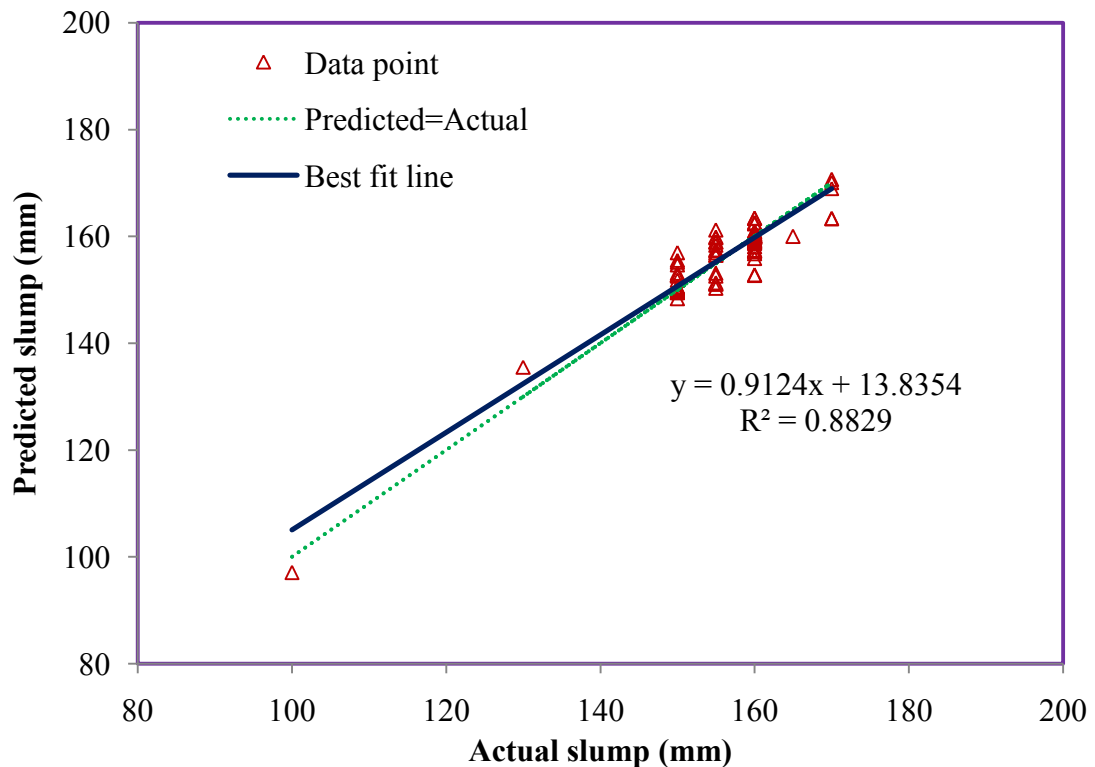
The results of the performance metrics for second RMC plant are shown in **Table 5.17**.

**Table 5.17:** Model performance metrics for the data set collected from second RMC plant

Model	RMSE (mm)	MAPE (%)	R	E	RSR	PBIAS (%)
GA-LMBNN	2.7646	1.2410	0.9396	0.8817	0.3436	0.0663

A comparison of the prediction performances shown above and in **Table 5.10** reveal that, the GA-LMBNN model does not yield the same accuracy of prediction for concrete slump for the design mix data collected from the second RMC batching plant.

A regression plot is drawn between the actual slump value and slump value predicted by the GA-LMBNN model for the second RMC plant. The regression plot is shown in **Figure 5.39**.



**Figure 5.39:** Regression plot between actual and model predicted slump for second RMC batching plant

To determine the existence of a significant linear relationship between the actual slump and model predicted slump value, following hypothesis for slope  $\beta_1$  of best fit line is tested:

**(a) The slope  $\beta_1$  is equal to unity**

$$H_0 : \beta_1 = 1 \quad (5.8)$$

$$H_a : \beta_1 \neq 1 \quad (5.9)$$

For the above hypothesis, the regression parameter estimates for slope  $\beta_1$ , t-values and p-value at 0.05 level of significance ( $\alpha$ ) for GA-LMBNN model are shown in **Table 5.18**.

**Table 5.18:** Regression parameter estimates, t-value and p-value for slope

Model	Slope $\beta_1$	Standard error	t-value	p-value
GA-LMBNN	0.9124	0.0336	2.6102	0.0105

The results exhibited above, show that for GA-LMBNN model, the t-value is greater than the critical value of t (= 1.9845) at degree of freedom (= 98) and the p-value is shown to be less than the level of significance  $\alpha$  (= 0.05). Hence, the null hypothesis is rejected, indicating that the linear relation between actual slump values and the slump values predicted by the GA-LMBNN model for the second RMC batching plant is not significant.

Following hypothesis for intercept  $\beta_0$  is also tested to assess closeness of best-fit regression line and line of equality (Actual=Predicted line):

**(a) Intercept  $\beta_0$  is equal to zero**

$$H_0 : \beta_0 = 0 \quad (5.10)$$

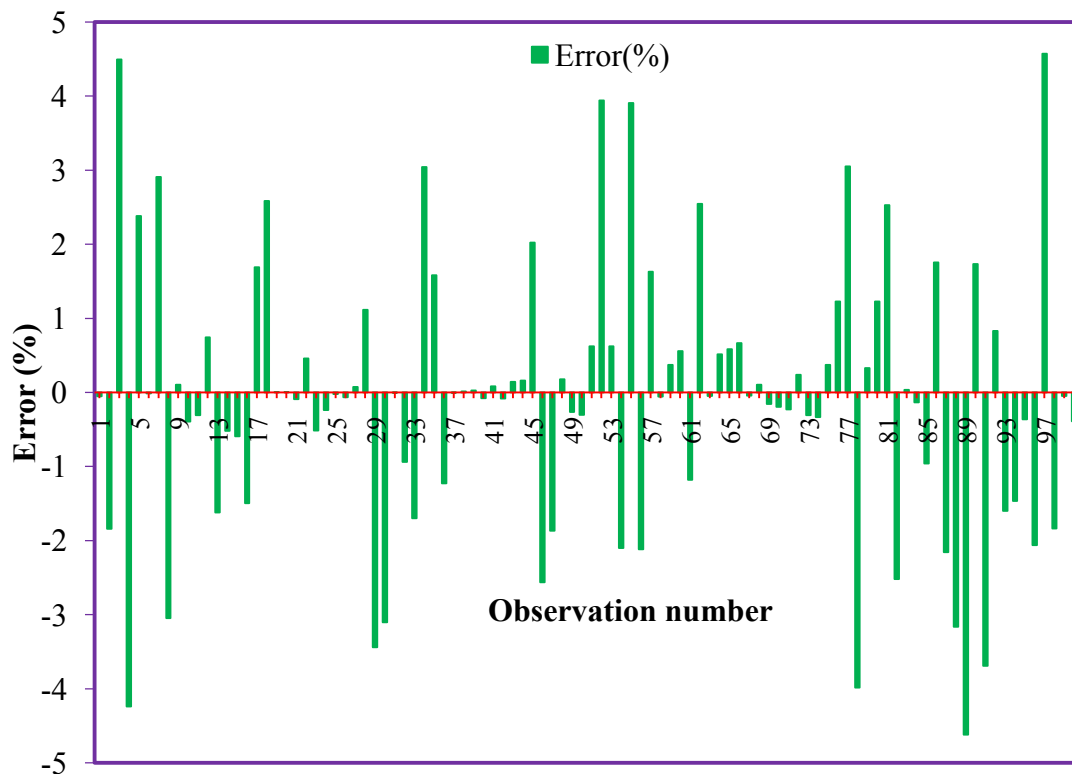
$$H_a : \beta_0 \neq 0 \quad (5.11)$$

For the above hypothesis, the regression parameter estimates for intercept  $\beta_0$ , t-values and p-value at 0.05 level of significance ( $\alpha$ ) for GA-LMBNN model are shown in **Table 5.19**.

**Table 5.19:** Regression parameter estimates, t-value and p-value for intercept

Model	Intercept $\beta_0$	Standard error	t-value	p-value
GA-LMBNN	13.8354	5.2675	2.6265	0.0100

The results exhibited above, show that for GA-LMBNN model, the t-value is greater than the critical value of t (= 1.9845) at degree of freedom (= 98) and the p-value is shown to be less than the level of significance  $\alpha$  (= 0.05). Hence, the null hypothesis is rejected indicating that, the intercept is significant. The best-fit regression line is shown to be placed slightly away from the line of equality. The results indicate that, since almost all RMC batching plants derive their raw material from different sources, any minor change in the properties of concrete constituents has a significant effect on the resulting slump value.

**Figure 5.40:** Percentage error for the data collected from second RMC batching plant

However, in **Figure 5.40** it is shown that, the developed hybrid GA-LMBNN model for the concrete slump can reliably predict the slump values within  $\pm 5\%$  error range for the second RMC batching plant indicating that, in case of RMC plants located in a particular area or city, that have a potential likelihood to derive almost all

or some of the raw material from the same source, the slump model developed for a particular RMC plant can be utilized for providing a approximate estimate of concrete slump values for design mix proportions of different RMC batching plants

## 5.9 Decision support tool to estimate slump for the RMC design mix proportions

In the preceding sections, it has been shown that the hybrid GA-LMBNN model for slump prediction yielded the best prediction accuracy and reliability. The GA-LMBNN model was saved and utilized to develop a decision support tool to aid quick and fair estimation of concrete slump for the user specified proportions of design mix constituents. The tool for estimating the slump value of RMC is shown in **Figure 5.41**.

**ESTIMATION OF SLUMP VALUE FOR READY MIX CONCRETE**

<u>Design Mix Constituents</u>	<u>Specific Gravity</u>	<u>Design Mix proportions</u>		<u>Design Mix ratios</u>	<u>Estimated Slump</u>
CEMENT	3.15	• [ ] •	330 kg/m <sup>3</sup>	Superplasticizer to binder 0.009	<b>159.691</b> mm
PFA	2.2	• [ ] •	100 kg/m <sup>3</sup>	Water to binder 0.3953	
SAND	2.66	• [ ] •	670 kg/m <sup>3</sup>	Fine to coarse aggregate 0.6091	
CA(20 mm)	2.65	• [ ] •	560 kg/m <sup>3</sup>	PFA to binder 0.2326	
CA(10 mm)	2.65	• [ ] •	540 kg/m <sup>3</sup>	Binder to aggregate 0.2429	
SUPERPLASTICIZER	1.2	• [ ] •	4.0 kg/m <sup>3</sup>		
WATER	1	• [ ] •	170 kg/m <sup>3</sup>		
Total Volume			2374 kg/m <sup>3</sup> 0.9905 m <sup>3</sup>	<input type="button" value="COMPUTE SLUMP"/>	

**Figure 5.41:** Decision support tool to estimate slump value for RMC

The user can specify the specific gravity of the design mix constituents and can alter the design mix proportions. The design mix ratios and volume of the design mix can also be monitored using this tool. After entering all input data namely, specific gravity and design mix proportions, the user is required to click the “COMPUTE SLUMP” button to evaluate the slump value based on the developed slump model. The tool will enable quick determination of slump value without performing the cumbersome and time-consuming slump test at the RMC batching plant. The tool gives the technical personnel in charge of design mix the liberty to experiment with numerous design mix proportions for designing a concrete mix satisfying the desired workability.

### 5.10 Summary

A three layered Feedforward Neural Network model was constructed for modeling slump of RMC. The model comprised of seven input neurons signifying the proportions of design mix constituents and one output neuron representing the concrete slump. A series of neural network architectures were trained using Levenberg-Marquardt (LM) backpropagation algorithm and examined to determine the optimal number of hidden layer neurons. The neural network model with eleven hidden layer neurons is shown to provide a balance of both learning and generalization and adopted for the further study.

To propose the hybrid GA-LMBNN model for the concrete slump, several parametric investigations were carried out to find the optimum GA parameters namely, population size, crossover fraction, and mutation rate. It is shown that a population of 40 chromosomes, crossover fraction 0.80 and mutation rate 0.015 yielded the best combination of parameters for GA.

The optimal weights and biases were evolved using GA assisted training of MFNN. The GA model was run twenty times, and it was found that the GA model took an average 13.7189 seconds and 39 population generations to train the MFNN. At the initial stage, the GA is shown to efficiently train the MFNN. As the training progresses and the search space are narrowed down, the convergence is adversely affected due to GA's weak local search ability. However, LM algorithm initialized with GA evolved weights and biases is shown to render quick convergence to the MFNN.

The hybrid GA-LMBNN model was initialized using GA evolved weights and biases that were further fine-tuned using LM algorithm. On the other hand, conventional LMBNN was initialized using randomly generated weights and biases in the range [-0.5, 0.5]. A comparison between GA-LMBNN and conventional LMBNN model showed that, the hybridization of GA and LMBNN leads to an approximate 80% reduction in training epochs and CPU time and reduces the stabilizes the training process.

The GA model yielded higher RMSE and MAPE values, lower R value and unacceptable E and RSR values, indicating its effectiveness to train MFNN. A lack of consistency in the prediction accuracy of the LMBNN model is noticeable, which is

shown to be more pronounced during its generalization stage when presented with the validation dataset. However, hybridization of GA with LMBNN during its training phase stabilizes the performance of the neural network model, augmenting and providing consistency to its learning and generalization accuracy. In nutshell based on the results, the hybrid GA-LMBNN is shown to outperform the GA and the conventional LMBNN model.

The first order and second regression models for slump were also developed. Compared to the first order regression model, a significant improvement in the prediction performance is noticeable in the case of the second order regression model. A comparison of model prediction error shows that neural network models yield lesser skewness and a smaller range of errors than the regression models. The hybrid GA-LMBNN model is shown to provide minimum skewness of prediction errors and maximum frequency of errors close to zero.

The results of the performance metrics established the prediction superiority of the neural network models in comparison to the conventional regression models. Amongst the concrete slump models, the GA-LMBNN model is shown to exhibit superior accuracy with lower RMSE, MAPE and RSR values along with, higher R and E values. The GA-LMBNN model is also shown to exhibit a very low PBIAS value indicating near to optimal prediction accuracy.

The percent correct plots reveal that, although regression models predict approximately 90% data within  $\pm 5\%$  threshold ranges, they are found to be inefficient at lower threshold error ranges. Compared to the LMBNN model, that is shown to predict 98% of the data correctly in the  $\pm 5\%$  error range, the GA-LMBNN model is shown to predict 100% of the data reliably at  $\pm 4\%$  error range and is found to be efficient at the smaller error ranges. The results exhibit the prediction accuracy and reliability of the neural networks and demonstrate their potential for modeling unstructured problems governed by unknown or complex functional relationships.

The hypothesis test for the regression plots reveal that, although all models developed for the concrete slump provided a linear relationship between the actual slump and slump values predicted by them but, the hybrid GA-LMBNN model was shown to provide the highest prediction accuracy. The neural network models namely, LMBNN and GA-LMBNN are shown to provide better  $R^2$  values in comparison to regression models, demonstrating their potential to accurately model the complex

interactions between the proportions of concrete design mix constituents and the slump value.

The relative importance of various constituents on slump value showed that superplasticizer and PFA are the most significant design mix constituents for RMC slump value. The response trace plots provided insight into the complex material behavior exhibited by the concrete slump. The plots indicate that there exists a certain level of each constituent beyond which a significant change in the behavior of concrete slump is noticeable.

The developed GA-LMBNN model for concrete slump does not yield the same prediction accuracy when presented with proportions of design mix constituents collected from a different RMC batching plant. However, it is shown that the hybrid GA-LMBNN model can reliably predict the slump values within  $\pm 5\%$  error range for design mix proportions of different RMC batching plant. The hybrid GA-LMBNN model is harnessed to develop a decision support tool for quickly predicting slump values without performing the cumbersome and time-consuming slump test at the RMC batching plant. The tool gives the technical personnel in charge of design mix liberty, to experiment with numerous design mix proportions for designing a concrete mix having customized slump value.

## **Chapter 6**

### **Summary and Conclusions**

---

- **Research summary**
  - **Conclusions**
  - **Recommendations**
  - **Future scope of study**
-



## 6.1 Research summary

Workability is an important property of fresh concrete. It is defined as the mechanical work required to manipulate the concrete mix with minimum loss of homogeneity. Since workability indicates the internal work needed for placing and compaction of fresh concrete and depends on the type and method of construction hence, there no single well-accepted test for measuring workability. However, in quantitative terms, the workability of concrete is measured using a widely used test called the Slump Test. Assessment of workability in terms of slump value plays a significant role in the Ready Mix Concrete (RMC) industry, as it not only controls quality and uniformity of concrete from batch to batch but also acts a measure to ensure that RMC transported with long delivery times is still in the state that it could be easily placed, compacted and finished at the construction site.

The constituents of RMC namely, cement, pulverized fuel ash, fine aggregate, coarse aggregate, superplasticizer and water, each exhibiting different physical and chemical properties, impart complexity to its material behavior, necessitating a non-algorithmic approach to its material modeling. To study, model and analyze such problems, approximate computer based Soft Computing techniques inspired by the reasoning, intuition, consciousness and wisdom possessed by human beings are employed. Amongst the various Soft Computing methodologies, the Genetic Algorithms (GA) and Artificial Neural Networks (ANN) have found wide applicability as optimization tool and modeling tool respectively. A review of the literature shows that the backpropagation (BP) algorithm trained Multilayer Feedforward Neural Networks (MFNN) also known as backpropagation neural networks (BPNN) are good function approximators. They have been extensively harnessed for tasks associated with modeling of physical phenomenon and material behavior, wherein conventional regression models do not yield the desired accuracy and predictability. Moreover, in all previous applications, the researchers have harnessed the BPNN for modeling the slump of concrete.

The reason for the wide applicability of BPNN is due to its ability to imbibe the non-linear or unknown input-output relationships easily through systematic updating of neural network weights and biases. Despite its wide popularity and simple implementation, the BP algorithm is faced with some inherent drawbacks. BP algorithm is a local search algorithm that harnesses the principle of steepest descent

and therefore it is highly dependent on the initial weights and biases to escape local minima and accomplish rapid convergence. During past few decades, several improved versions of local search BP algorithms have been introduced. Amongst them, the Levenberg-Marquardt (LM) backpropagation algorithm is shown to provide both learning efficiency and fast convergence. Although the LM backpropagation algorithm inherits fast convergence of the Gauss-Newton algorithm yet, it still carries the weakness of the steepest descent algorithm. Genetic Algorithms (GA) on the other hand are global search algorithms that employ non-trajectory search to perform longer jumps in search space. Although, gradient-free search employed by GA minimizes the probability of entrapment at local minima yet, owing to its weak local search ability, it is susceptible to slow convergence.

To circumvent the inherent drawbacks of GA and BP algorithms, a hybrid methodology is proposed for modeling the relationship between slump of concrete and the proportions of RMC design mix constituents. The methodology amalgamates the global search ability of GA with the fast converging LM backpropagation algorithm, for assisting the MFNN to escape local minima and accomplish faster convergence. The study further attempts to (a) assess and compare the effectiveness of Genetic Algorithms, Levenberg-Marquardt backpropagation algorithm and hybrid Genetic Algorithm-Levenberg Marquardt backpropagation algorithm for training the MFNN; (b) compare the prediction accuracy and reliability of the neural network and regression models; (c) analyze and explore the material behaviour of concrete slump; (d) assess the effectiveness and applicability of the slump model for a different RMC batching plant; and (e) develop a decision support tool to estimate the slump value for the concrete design mix.

The data for the study were collected from two different RMC batching plants. In all a total of 493 and 100 design mixes of concrete grade M10 to M35 were collected from the first and the second RMC batching plant respectively. The data constituted the design mix proportions of cement, pulverized fuel ash or fly ash (PFA), sand, coarse aggregate (20mm), coarse aggregate (10mm), superplasticizer and water in  $\text{kg/m}^3$ , along with their corresponding slump value in mm. The entire data is randomized and split into three statistically similar subsets namely, training dataset, validation dataset, and test dataset. The data was normalized in the range [-1, 1] to minimize the possibility of bias towards a particular feature. The Neural Network

Toolbox and Global Optimization Toolbox included in the commercially available software MATLAB 7.1 (R14 SP3) (Version 7.1.0.246) were used to implement BPNN and GA respectively.

The three layer architecture of MFNN adopted for the study comprised of “input layer”, an “output layer” and a sandwiched “hidden layer” containing artificial neurons, permitting only inter-layer connections among the artificial neurons in the forward direction only. The model used the seven neurons representing design mix proportions of RMC as inputs to predict the slump value of concrete. The MFNN model is trained using Levenberg-Marquardt backpropagation algorithm and is denoted as LMBNN.

The hybrid GA-LMBNN methodology comprised of two stages. In the first stage, the GA was employed to train the MFNN, for evolving the optimal weights and biases for the MFNN. During the second stage, the GA evolved weights and biases were used for initializing LM backpropagation algorithm. The LM algorithm starts from the optimized set of weights and biases and fine tune them to train the MFNN model. The conventional LMBNN model, on the other hand, was initialized with the random value of weights and biases in the range [-0.5, 0.5].

Regression is a highly useful statistical technique for developing a quantitative relationship between the dependent variable and one or more independent variable. In the present study, there are seven independent variables representing the design mix constituents of RMC and one dependent variable signifying the slump value. The regression models namely, first order and second order models were developed for establishing a relationship between the proportions of the design mix constituents of RMC and the slump value.

Six different statistical performance metrics namely, root mean square error (RMSE), mean absolute percentage error (MAPE), correlation coefficient (R), coefficient of efficiency (E), root mean square error to the observation’s standard deviation ratio (RSR) and percent bias error (PBIAS) have been combined to derive holistic inferences regarding the prediction performance of the slump models.

In the present study the sensitivity analysis was performed using connection weights method and response trace plots. The importance of each design mix constituent of RMC on the slump value was assessed using connection weights

method. The response on the slump value brought about by varying the proportions of each design mix constituent was deduced using the response trace plots. The effectiveness and universal applicability of the slump model were assessed by, utilizing it to predict the slump for concrete design mix data collected from the other RMC batching plant. The knowledge extracted from RMC data in the form of a neural network model was used to develop a decision support tool for estimating initial slump for the concrete design mix.

A trial and error technique was employed for determining the optimal number of hidden layer neurons, for which a series of neural network architectures were examined. The LM backpropagation algorithm is employed to train the MFNN model using the training data set. The validation dataset is used to monitor the average standard deviation of errors (AVSD) and average of mean absolute error (AVMAE) at each training epoch to avoid the overfitting of the model. The optimal architecture for the MFNN was shown to comprise eleven hidden layer neurons and denoted as 7-11-1.

GA is a population-based heuristic that employs computational models of evolutionary processes like selection, crossover, and mutation wherein, the solutions to the problems are encoded as genes of the chromosome population. In the study, the weights and biases for the MFNN architecture (7-11-1) numbering 100, are encoded as genes of the chromosomes. The fitness of each chromosome is measured using root mean square error (RMSE) between the actual and the predicted slump value. Tournament selection strategy is adopted for allowing the chromosomes of higher fitness to pass on their genes to the next generation while prohibiting the entrance of low fitness chromosomes. Several parametric investigations were carried out to find the optimum value for the most influential GA parameters namely, population size, crossover fraction, and mutation rate. An optimum value of initial population size of 40 chromosomes, scattered crossover fraction 0.80 and uniform mutation rate 0.015 is adopted for the present study.

The optimal initial weights and biases for the MFNN architecture (7-11-1) are determined using GA. The GA took an average CPU time of 13.7189 seconds and 39 population generations to evolve the optimal weights and biases. The training plot of GA assisted training of MFNN reveals that the convergence of GA is adversely affected as the search space is narrowed down, indicating its weak local search ability.

A comparison of hybrid GA-LMBNN initialized with GA evolved optimal weights and biases with conventional LMBNN model initialized with randomly generated weights and biases in the range [-0.5, 0.5] reveal that the hybrid GA-LMBNN is more stable and exhibits faster convergence. The hybrid GA-LMBNN model is shown provide an approximate 80% reduction in training epochs and CPU time.

The statistical performance metrics show that the GA model gave a relatively poor learning and generalization accuracy in comparison to the neural network models namely, GA-LMBNN and LMBNN, indicating its ineffectiveness to train the MFNN. A lack of consistency in the prediction accuracy of the LMBNN model is noticeable, which is shown to be more pronounced during its generalization stage when presented with the validation dataset. However, hybridization of GA with LMBNN during its training phase stabilizes the performance of the neural network model, augmenting and providing consistency to its learning and generalization accuracy. In nutshell based on the results, the hybrid GA-LMBNN is shown to outperform the GA and the conventional LMBNN model.

In comparison to first order and second order regression models for the concrete slump, the neural network models namely, LMBNN and hybrid GA-LMBNN model exhibited a smaller range of model errors and better prediction reliability and accuracy. The hypothesis test for the regression plots reveal that the hybrid GA-LMBNN model can predict the values of slump very close to the actual values, proving its effectiveness to model the complex interaction between the proportions of concrete design mix constituents and the slump value.

The superplasticizer and PFA in concrete were shown to exhibit maximum positive influence on the slump value of concrete. Whereas, the coarse aggregate 10 mm is shown to exhibit a maximum negative influence on the concrete slump. The study showed that superplasticizer and PFA are the most important design mix constituents for RMC slump value. The response trace plots graphically exhibited the effect of an increase in each constituent of concrete on the slump value. The plots indicate that there exists a certain critical level of each constituent beyond which a significant change in the behavior of concrete slump is noticeable.

The hybrid GA-LMBNN model for slump does not yield the same prediction accuracy when presented with proportions of design mix constituents collected from a different RMC batching plant proving that, the slump not only depends on the

proportions but also on properties of the design mix constituents of concrete. However, the hybrid GA-LMBNN is shown to predict the slump values reliably within  $\pm 5\%$  error range. The decision support tool is shown to quickly and reliably predict slump based on the design mix proportions and can be utilized to experiment with numerous design mix proportions for formulating concrete design mixes catering to the workability requirements without performing actual slump tests at the RMC batching plant.

## 6.2 Conclusions

The broad conclusions derived from the study are as under:

1. The effectiveness of Multilayer Feedforward Neural Network (MFNN) training algorithms namely, Genetic Algorithms (GA), Levenberg-Marquardt (LM) backpropagation algorithm and hybrid GA-LM algorithm, discussed and examined in the study reveal that:
  - (i) The convergence speed of GA is severely affected in the vicinity of global optimum owing to its weak local search capability, as it requires much computational effort to reduce the network error to a minimum acceptable value. As a consequence, the MFNN trained using GA is shown to exhibit poor learning and generalization performance. The global search algorithms like Genetic Algorithms cannot be, therefore, regarded as an efficient alternative to the local search backpropagation algorithms for training the MFNN.
  - (ii) The hybridization of GA with Levenberg-Marquardt (LM) backpropagation algorithm helped to derive the best from the global search ability of GA and the fast local search rendered by LM algorithm, thereby offering dual benefits:
    - (a) Firstly, the non-trajectory multidimensional search rendered by GA facilitates the LM algorithm to reach a global solution to a problem by narrowing down and restricting its search in the regions where there are paramount possibilities of finding the optimal neural network weights and biases. Although, on account

of the computational effort taken by GA to evolve optimal initial weight and biases for the LM algorithm, the hybrid GA-LM algorithm is found to be approximately 10 to 15 times computationally expensive than the conventional LM algorithm yet, it is shown to circumvent the inherent drawback of the conventional LM algorithm getting trapped in local minima. This advantage is reflected by the consistent and improved learning and generalization performance of the hybrid GA-LMBNN model compared to the conventional LMBNN model that uses randomly initialized weights and biases.

- (b) Secondly, once the probability of getting trapped in local minima is minimized and the search region is narrowed, the Gauss-Newton algorithm present in the LM algorithm augments the convergence speed, causing 80% reduction in the average CPU time and epochs taken by the conventional LM algorithm to train the MFNN.
2. A comparison of the prediction performance and reliability of the artificial neural networks and regression models for the concrete slump show that:
- (i) The error histograms for the first order and second order regression models exhibit skewness and a large range of prediction errors. On the other hand, the prediction errors of neural network models are confined to a smaller range and exhibit higher frequency of errors close to zero with smaller skewness value. The hybrid methodology harnessing GA and LMBNN is shown to render symmetry to the distribution of errors and increase the frequency of errors close to zero.
  - (ii) The neural network models namely GA-LMBNN and LMBNN yielded a lower RMSE, MAPE and RSR values and higher R and E values, indicating a significantly improved performance accuracy in comparison to the first order and second order regression models. Among the concrete slump models, the GA-LMBNN model is shown to exhibit superior accuracy with lower RMSE, MAPE and RSR

values of 0.8015 mm, 0.3197% and 0.0975 respectively along with, higher R and E values of 0.9952 and 0.9905 respectively. The GA-LMBNN model is also shown to exhibit a very low PBIAS value of 0.0208% indicating near to optimal prediction accuracy.

- (iii) The regression models were able to predict correctly nearly 90% of the slump values within  $\pm 5\%$  error range. On the other hand, the neural network models are shown to predict correctly 98% of the slump values within  $\pm 5\%$  error range and exhibit significant efficiency at smaller threshold error ranges. The hybrid GA-LMBNN model is shown to predict correctly 100% slump values within  $\pm 4\%$  error range and, therefore, outperform the prediction accuracy of the conventional LMBNN model. The results exhibit the prediction accuracy and reliability of the neural networks and demonstrate their potential for modeling unstructured problems governed by unknown or complex functional relationships.
  - (iv) The hypothesis test for the regression equation between actual and model predicted slump values reveal that, the first order regression, the second order regression and the conventional LMBNN models developed for concrete slump are able to provide a fair estimate of the concrete slump based on the proportions of the design mix constituents. However, the hybrid GA-LMBNN model is shown to provide significant prediction accuracy thereby, outperforming the regression models and conventional LMBNN model. The regression equation is shown to fit 42.85%, 73.48%, 95.90% and 99.05% data for the first order regression, second order regression, LMBNN, and GA-LMBNN models respectively indicating that, unlike in the case of the neural network models, the degree of correlation between the actual and the predicted slump of concrete in case of the regression models is not very strong. It can, therefore, be inferred that, like other properties of concrete, the slump also bears a highly non-linear functional relationship with the design mix constituents.
3. The trained hybrid GA-LMBNN model is shown to extract significant information regarding the complex and unknown interaction between the



concrete's design mix constituents and its physical property, i.e., workability measured using the slump value. The connection weights method and response trace plots were harnessed to provide insight into the complex material behavior exhibited by the concrete slump. The conclusions drawn from sensitivity analysis are as under:

- (i) The constituents of the RMC namely, superplasticizer (SP) and pulverized fuel ash (PFA) are shown to exhibit positive influence on the concrete slump and together contribute a significant 52% importance, indicating their significance in the RMC industry. Since water and cement are used as a ratio in the design mix and is almost fixed for a customized grade of concrete, these two constituents have exhibited almost the equal importance on the concrete slump. A predominant amount of undersized particles aggregate dust, silt and murrum that inadvertently get mixed with CA (10mm) is shown to cause a detrimental effect on the initial slump of RMC. On the other hand, due to the presence of larger-sized aggregates in CA (20mm), it is shown to exhibit a positive influence on the slump value. Sand quarried from different natural sources causes deviation in the gradation of particles, increasing particle interference and surface area leading to decrease in slump value. The findings of the study supported by grading analysis of CA (10mm) and sand obtained from RMC industry reveals that, the workability of concrete is sensitive to the grading of the aggregates.
- (ii) The response trace plots helped to explore the complex non-linear material behavior exhibited by the concrete slump. Response trace plots for cement, PFA and water have shown a sharp rise in the concrete slump. However, beyond a certain level of these constituents, the slump falls sharply. The response plot for superplasticizer is seen to rise sharply, however, beyond particular superplasticizer content, it ceases to improve the slump value further. On the other hand, the sand and CA (20mm) exhibited a flatter response, and CA (10mm) is shown to follow a declining trend. The response trace plots have shown that, there exists a certain critical level of design mix

proportion beyond which a significant change in the behavior of concrete slump is noticeable.

4. The hybrid GA-LMBNN model for the slump is shown to predict the slump values accurately within  $\pm 5\%$  error range for design mix proportions collected from a different RMC batching plant. However, in contrast to the first RMC batching plant, the GA-LMBNN model does not yield the same degree of prediction accuracy for the second RMC batching plant. It indicates that, the concrete slump is sensitive to the physical and chemical properties of the individual design mix constituents, and therefore, any minor change in the properties of the constituent can significantly affect the prediction accuracy of the slump model. The concrete slump is thus, shown to depend not only on the proportions but also on the properties of the individual design mix constituents. The slump model developed using data collected from a particular RMC batching plant can, therefore, be reliably and universally applied for estimation of concrete slump for different RMC batching plants as long as these plants derive their raw material from the same source.
5. The decision support tool harnessing the hybrid GA-LMBNN model for concrete slump will assist the technical person in charge of the concrete design mix, to quickly estimate the initial slump of concrete at the batching plant to a desired degree of accuracy based on the design mix proportions, without undertaking the cumbersome experimental procedures. The decision support tool provides the liberty to the technical personnel to experiment with numerous concrete mix proportions for designing a customized RMC mix that can render slump value at the construction site within the specified range, even after suffering slump loss during its transit from the batching plant to the construction site.

### 6.3 Recommendations

Based on the study following recommendations are made:

- a) The hybridization of artificial neural networks with heuristic global search algorithms can significantly improve the prediction accuracy of the neural

network model. The neural network model based on the hybrid methodology can, therefore, render accurate modeling for problems that are unstructured and highly complex in nature.

- b) The weights of the neural network model have been utilized to extract the relative importance of various design mix constituents on the slump value. Based on the results, the raw material suppliers for RMC plants can be guided accordingly to improve the quality of the material to minimize the negative effect rendered by some design mix constituents on the slump value. For example, by adopting latest technology like the three stage vertical shaft impact crusher (VSI), an improvement in the quality of the coarse aggregates can be achieved.
- c) An effective knowledge sharing and exchange of experience between the academicians and the industry personnel to improve the properties of the RMC can be achieved using this study.
- d) The neural network model has provided insight into the complex material behavior of concrete. The methodology can be used to analyze and compare the behavior of concrete manufactured using different source materials.

#### **6.4 Future scope of study**

In the present study, GA has been hybridized with the fastest converging Levenberg-Marquardt backpropagation algorithm for training the Feedforward Neural Network. Recently, an updated version of Feedforward Neural Network with a single layer of hidden neurons called the Extreme Learning Machines (ELM) has been developed to render fast convergence coupled with good generalization. However, in ELM, the hidden layer neurons and values of weights and biases are randomly chosen, which can significantly affect the performance of the trained model. GA hybridized with ELM for evolving the initial weights and biases and hidden layer neurons, can be used in the future study for quick and accurate material modeling of concrete.

The scope of the present study was limited to model concrete slump based on the design mix proportions collected from an RMC batching plant. Along with the slump data, the compressive strength data for the design mix proportions can be

collected from the RMC batching plant for developing a neural network model that can simultaneously predict slump value along with the compressive strength for the design mix proportions. The technical personnel in charge of mix design can use the comprehensive model as a tool for quick determination of compressive strength and slump value and can be further harnessed to provide insight into the complex material behavior of concrete. The proportions of the design mix for developing the slump model in the study were limited to M35 grade concrete. The methodology presented in the study can be extended and applied to modeling the compressive strength and slump of higher grade concretes.

## **References**

- Abdollahzadeh, A., Masoudnia, R. and Aghababaei, S. (2011) 'Predict strength of rubberized concrete using artificial neural network', *WSEAS Transactions on Computers*, 2(10), pp. 31–40.
- Abdullah, S.S., Malek, M.A., Mustapha, A. and Aryanfar, A. (2014) 'Hybrid of artificial neural network-genetic algorithm for prediction of reference evapotranspiration ( $ET_0$ ) in arid and semiarid regions', *Journal of Agricultural Science*, 6(3), pp. 191–200.
- Acciani, G., Chiarantoni, E. and Fornarelli, G. (2006) 'A neural network approach to study  $O_3$  and  $PM_{10}$  concentration', *Proceedings of the 16<sup>th</sup> International Conference on Artificial Neural Networks (ICANN'06)*. Berlin, Germany. Springer-Verlag, pp. 913–922.
- Adamowski, J., Chan, H.F., Prasher, S.O., Ozga-Zielinski, B. and Sliusarieva, A. (2012) 'Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada', *Water Resources Research*, 48(1), pp.1–14.
- Ai, H. and Guo, S. (2014) 'Bridge Health Evaluation System based on the Optimal BP Neural Network', *International Journal of Control and Automation*, 7(1), pp. 331–338.
- Albuquerque, A.T., El Debs, M.K. and Melo, A.M.C. (2012) 'A cost optimization-based design of precast concrete floors using genetic algorithms', *Automation in Construction*, 22, pp. 348–356.
- Alshihri, M.M., Azmy, A.M. and El-Bisy, M.S. (2009) 'Neural networks for predicting compressive strength of structural light weight concrete', *Construction and Building Materials*, 23(6), pp.2214–2219.
- Amin, A.E. (2013) 'A novel classification model for cotton yarn quality based on trained neural network using genetic algorithm', *Knowledge-Based Systems*, 39, pp. 124–132.
- Arbib, M. (1995) *The handbook of brain theory and neural networks*. Cambridge Mass: MIT Press.
- Asadi, S., Hadavandi, E., Mehmanpazir, F. and Nakhostin, M.M. (2012) 'Predictability and forecasting automotive price based on a hybrid train algorithm of MLP neural network', *Knowledge-Based Systems*, 35, pp. 245–258.
- Asadi, S., Shahrabi, J., Abbaszadeh, P. and Tabanmehr, S. (2013) 'A new hybrid artificial neural networks for rainfall-runoff process modeling', *Neurocomputing*, 121, pp. 470–480.
- Atici, U. (2011) 'Prediction of the strength of mineral admixture concrete using multivariable regression analysis and an artificial neural network', *Expert Systems with Applications*, 38(8), pp. 9609–9618.

- Attoh-Okine, N. (2004) 'Application of genetic-based neural network to lateritic soil strength modeling', *Construction and Building Materials*, 18(8), pp. 619–623.
- Avila, C., Shiraishi, Y. and Tsuji, Y. (2004) 'Crack width prediction of reinforced concrete structures by artificial neural networks', *Proceedings of the 7<sup>th</sup> Seminar on Neural Network Applications in Electrical Engineering, 2004: NEUREL 2004*. Belgrade, Serbia. 23<sup>rd</sup> -25<sup>th</sup> September 2004. IEEE, pp. 39–44.
- Aydin, Z. and Ayvaz, Y. (2013) 'Overall cost optimization of prestressed concrete bridge using genetic algorithm', *KSCE Journal of Civil Engineering*, 17(4), pp. 769–776.
- Baghalian, S. and Nazari, F. (2011) 'Prediction of uplift pressure under the diversion dam using artificial neural network and genetic algorithm', *International Journal of Engineering & Applied Sciences*, 3(3), pp. 23–32.
- Bagheri, M., Mirbagheri, S.A., Bagheri, Z. and Kamarkhani, A.M. (2015) 'An Artificial Neural Network based expert system fitted with Genetic Algorithms for detecting the status of several rotary components in agro-industrial machines using a single vibration signal', *Process Safety and Environmental Protection*, 95, pp. 12–25.
- Bai, J., Wild, S., Ware, J.A. and Sabir, A.A. (2003) 'Using neural networks to predict workability of concrete incorporating metakaolin and fly ash', *Advances in Engineering Software*, 34(11-12), pp. 663–669.
- Bal, L. and Buyle-Bodin, F. (2014) 'Artificial neural network for predicting creep of concrete', *Neural Computing and Applications*, 25(6), pp. 1359–1367.
- Basheer, I.A. and Hajmeer, M. (2000). 'Artificial neural networks: fundamentals, computing, design, and application', *Journal of Microbiological Methods*, 43(1), pp. 3–31.
- Ben-Romdhane, H., Alba, E. and Krichen, S. (2013) 'Best practices in measuring algorithm performance for dynamic optimization problems', *Soft Computing*, 17(6), pp. 1005–1017.
- Bertagnoli, G., Giordano, L. and Mancini, S. (2014) 'Optimization of concrete shells using genetic algorithms', *ZAMM-Journal of Applied Mathematics and Mechanics*, 94(1–2), pp. 43–54.
- Bilgil, A. (2012) 'Estimation of slump value and Bingham parameters of fresh concrete mixture composition with artificial neural network modelling', *Scientific Research and Essays*, 5(8), pp. 1753–1765.
- Bishop, C.M. (1995) *Neural networks for pattern recognition*. New York: Oxford University Press.
- Bostanci, S.C., Limbachiya, M. and Kew, H. (2016) 'Portland-composite and composite cement concrete made with coarse recycled glass sand aggregates: Engineering and durability properties', *Construction and Building Materials*, 128, pp. 324–340.

- Boukhatem, B., Kenai, S., Hamou, A.T., Ziou, D. and Ghrici, M. (2012) 'Predicting concrete properties using neural networks (NN) with principal component analysis (PCA) technique', *Computers and Concrete*, 10(6), pp. 557–573.
- Bowden, G.J., Maier, H.R. and Dandy, G.C. (2002) 'Optimal division of data for neural network models in water resources applications', *Water Resources Research*, 38(2), pp. 2-1 –2-11.
- Briki, L. and Djeghaba, K. (2015) 'A new plasticity model for concrete in compression based on Artificial Neural Networks', *International Journal of Advanced Science and Technology*, 75, pp. 43–50.
- Chandre Gowda C. and Mayya, S.G. (2014) 'Comparison of Back Propagation Neural Network and Genetic Algorithm Neural Network for Stream Flow Prediction', *Journal of Computational Environmental Sciences*, 2014, Article ID: 290127.
- Chen, G., Fu, K., Liang, Z., Sema, T., Li, C. and Tontiwachwuthikul, P. (2014). 'The genetic algorithm based back propagation neural network for MMP prediction in CO<sub>2</sub>-EOR process', *Fuel*, 126(15), pp. 202–212.
- Chindaprasirt, P., Buapa, N. and Cao, H.T. (2005). 'Mixed cement containing fly ash for masonry and plastering work', *Construction and Building Materials*, 19, pp. 612–618.
- Chine, W.-H, Chen, L., Hsu, H. -H. and Wang, T, -S. (2010) 'Modeling slump of concrete using artificial neural networks', *Proceedings of 2010 International Conference on Artificial Intelligence and Computational Intelligence*. Sanya, China. 23<sup>rd</sup>–24<sup>th</sup> October 2004. IEEE, pp. 236–239.
- Chithra, S., Senthil Kumar, S.R.R., Chinnaraju, K., and Alfin Ashmita, F. (2016) 'A comparative study on the compressive strength prediction models for High Performance Concrete containing nano silica and copper slag using regression analysis and Artificial Neural Networks', *Construction and Building Materials*, 114, pp. 528-535.
- Cybenko, G. (1989) 'Approximation by superposition of a sigmoidal function', *Mathematics of Control, Signals, and Systems*, 2, pp. 303–314.
- Daliakopoulos, I.N., Coulibaly, P. and Tsanis, I.K. (2005) 'Groundwater level forecasting using artificial neural networks', *Journal of Hydrology*, 309(1–4), pp. 229–240.
- Daniel, D.G. (2006) Factors Influencing Concrete Workability. In: Lamond, J.F. and Pielert, J.H. ed. *Significance of Tests and Properties of Concrete & Concrete-Making Materials*. West Conshohocken, PA. ASTM International, pp. 59–72.
- Dee, G.L., Bakhary, N., Rahman, A.A. and Ahmad, B.H. (2011) 'A comparison of artificial neural networks learning algorithms for vibration-based damage detection', *Advanced Materials Research*, 163–167, pp. 2756–2760.
- DeJong, K.A. and Spears, W.M. (1990). 'An analysis of the interacting roles of population size and crossover in genetic algorithms', *Proceedings of the First*



- Workshop on Parallel Problem Solving from Nature*. Springer-Verlag, Berlin. pp. 38–47.
- Despande, N., Londhe, S. and Kulkarni, S. (2014) ‘Modeling compressive strength of recycled aggregate concrete by Artificial Neural Network, Model Tree and Non-linear Regression’, *International Journal of Sustainable Built Environment*, 3(2), pp.187–198.
- Diab, A.M., Elyamany, H.E., Abd Elmoaty, M.A. and Shalan, A.H. (2014) ‘Prediction of concrete compressive strength due to long term sulfate attack using neural network’, *Alexandria Engineering Journal*, 53(3), pp. 627–642.
- Dias, W.P.S. and Pooliyadda, S.P. (2001) ‘Neural networks for predicting properties of concrete with admixtures’, *Construction and Building Materials*, 15(7), pp. 371–379.
- Ding, Y.R., Cai, Y.J., Sun, P.D. and Chen, B. (2014) ‘The use of combined neural networks and genetic algorithms for prediction of river water quality’, *Journal of Applied Research and Technology*, 12, pp. 493–499.
- Duan, Z. and Poon, C. (2014) ‘Factors affecting the properties of recycled concrete by using neural networks’, *Computers and Concrete*, 14(4), pp. 547–561.
- Dumne, S.M. (2014) ‘Effect of superplasticizer on Fresh and Hardened properties of Self-Compacting Concrete Containing Fly Ash’, *American Journal of Engineering Research*, 3(3), pp. 205–211.
- Erb, R.J. (1993). ‘Introduction to backpropagation neural network computation’, *Pharmaceutical Research*, 10(2), pp. 165–170.
- Feng, Y., Zhang, W., Sun, D. and Zhang, L. (2011) ‘Ozone concentration forecast method based on genetic algorithm optimized back propagation neural networks and support vector machine data classification’, *Atmospheric Environment*, 45(11), pp. 1979–1985.
- Flood, I. and Kartam, N. (1994) ‘Neural networks in civil engineering. I: Principles and understanding’, *Journal of Computing in Civil Engineering*, 8(2), pp. 131–148.
- Funahashi, K. (1989) ‘On approximate realization of continuous mappings by neural networks’, *Neural Networks*, 2(3), pp. 183–192.
- Gallahger, M. and Downs T. (1997) *Visualisation of learning in neural networks using principal component analysis*. In: Varma, B. and Yao, X., editors. Proceedings of international conference on computational intelligence and multimedia applications. Australia; 1997, pp. 327–331.
- Ghaffari, A., Abdollahi, H., Khoshayand, M.R., Bozchalooi, I.S., Dadgar, A. and Rafiee-Tehrani, M. (2006) ‘Performance comparison of neural network training algorithms in modeling of bimodal drug delivery’, *International Journal of Pharmaceutics*, 327, pp. 126–138.

- Ghafoori, N., Najimi, M., Sobhani, J. and Aqel, M.A. (2013) 'Predicting rapid chloride permeability of self-consolidating concrete: A comparative study on statistical and neural network models', *Construction and Building Materials*, 44, pp.381–390.
- Gomes, G. S. da S., Ludermir, T. B. and Lima, L. M. M. R. (2011) 'Comparison of new activation functions in neural network for forecasting financial time series', *Neural Computing and Applications*, 20(3), pp. 417–439.
- Gopala Krishna Sastry, K.V.S., Sudarsana Rao, H., Ramana Reddy, I.V. and Ghorpade, V.G. (2014) 'Development of Genetic Algorithm Based Macro Mechanical Model for Steel Fibre Reinforced Concrete', *International Journal of Engineering Research and Applications*, 4(1), pp. 231–237.
- Gorphade, V.G., Sudarsana Rao, H., and Beulah, R.M. (2013) 'Development of Genetic Algorithm based Neural Network Model for predicting strength of High Performance Concrete', *International Journal of Engineering Research and Applications (IJERA)*, 3(2), pp. 1687–1694.
- Grenfensette, J.J. (1986). 'Optimization of control parameters for genetic algorithms', *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), pp. 122–128.
- Gupta, H.V., Sorooshian, S. and Yapo, P.O. (1999) 'Status of automatic calibration for hydrologic models: Comparison with multilevel expert calibration', *Journal of Hydrologic Engineering*, 4(2), pp. 135–143.
- Hagan, M.T. and Menhaj, M.B. (1994). 'Training Feedforward Networks with the Marquardt Algorithm', *IEEE Transactions on Neural Networks*, 5(6), pp. 989–993.
- Haykin, S. (2009). *Neural Networks A Comprehensive Foundation*. India: Pearson Prentice Hall.
- Hodhod, O.A. and Ahmed, H.I. (2013) 'Developing an artificial neural network model to evaluate chloride diffusivity in high performance concrete', *HBRC Journal*, 9(1), pp. 15–21.
- Hola, J. and Schabowicz, K. (2005) 'Application of artificial neural networks to determine concrete compressive strength based on non-destructive tests', *Journal of Civil Engineering and Management*, 11(1), pp. 23–32.
- Hornik, K., Stinchcombe, M. and White, H. (1989) 'Multilayer feed forward networks are universal approximators', *Neural Networks*, 2(5), pp. 359–366.
- Huang G. and Wang, L. (2011) 'Hybrid Neural Network Models for Hydrologic Time Series Forecasting Based on Genetic Algorithm', *Proceedings of the Fourth International Joint Conference on Computational Sciences and Optimization (CSO)*. Yunnan, China. 15<sup>th</sup> -19<sup>th</sup> April 2011. IEEE, pp. 1347–1350.
- Hunter, D., Hao, Y., Pukish, M.S., Kolbusz, J. and Wilamowski, B.M. (2012) 'Selection of proper Neural Network sizes and architectures-A comparative study', *IEEE Transaction on Industrial Informatics*, 8(2), pp. 228–240.

- Ince, R. (2004) 'Prediction of fracture parameters of concrete by artificial neural networks', *Engineering Fracture Mechanics*, 71(15), pp. 2143–2159.
- IS: 383 (2016) 'Specifications for coarse and fine aggregates from natural sources for concrete', Bureau of Indian Standards.
- IS: 456 (2000) 'Plain and Reinforced Concrete-Code of Practice', Bureau of Indian Standards.
- Jain, A., Jha, S.K. and Misra, S. (2008) 'Modeling and analysis of concrete slump using artificial neural networks,' *Journal of Materials in Civil Engineering*, 20(9), pp. 628–633.
- Jalalkamali, A. and Jalalkamali N. (2011) 'Groundwater modeling using hybrid of artificial neural network with genetic algorithm', *African Journal of Agricultural Research*, 6(26), pp. 5774–5784.
- Jalili-Gazi Zade, M. and Noori, R. (2008) 'Prediction of municipal solid waste generation by use of artificial neural network: A case study of Mashhad', *International Journal of Environmental Research*, 2(1), pp. 13–22.
- Jianyong, L. and Yan, Y. (2001) 'A study on creep and drying shrinkage of high performance concrete', *Cement and Concrete Research*, 31, pp. 1203–1206.
- Jinchuan, K. and Xinzhe, L. (2008) 'Empirical Analysis of Optimal Hidden Neurons in Neural Network Modeling for Stock Prediction', *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application*. Wuhan, China. 19<sup>th</sup>-20<sup>th</sup> December 2008. IEEE, pp. 828–832.
- Johari, A., Javadi, A.A. and Habibagahi, G. (2011) 'Modelling the mechanical behaviour of unsaturated soils using a genetic algorithm-based neural network', *Computers and Geotechnics*, 38(1), pp. 2–13.
- Kalogirou, S.A. (2000) 'Applications of artificial neural-networks for energy systems', *Applied Energy*, 67, pp. 17–35.
- Kalogirou, S.A. (2003) 'Artificial intelligence for the modeling and control of combustion processes: A review', *Progress in Energy and Combustion Science*, 29(6), pp. 515–566.
- Kamp, R.G. and Savenije, H.H.G. (2006) 'Optimizing training data for ANNs using Genetic Algorithms', *Hydrology and Earth System Sciences*, 10, pp. 603–608.
- Karimi, H. and Yousefi, F. (2012) 'Application of artificial neural network–genetic algorithm (ANN–GA) to correlation of density in nanofluids', *Fluid Phase Equilibria*, 336, pp. 79–83.
- Karlik, B. and Olgac, A.V. (2011) 'Performance analysis of various activation functions in generalized MLP architectures of neural networks', *International Journal of Artificial Intelligence and Expert Systems (IJAE)*, 1(4), pp. 111–122.

- Karthikeyan, J., Upadhyay, A. and Bhandari, N.M. (2008) 'Artificial neural network for predicting creep and shrinkage of high performance concrete', *Journal of Advanced Concrete Technology*, 6(1), pp. 135–142.
- Kemp, S.J., Zaradic, P. and Hansen, F. (2007) 'An approach for determining relative input parameter importance and significance in artificial neural networks', *Ecological Modelling*, 204(3–4), pp. 326–334.
- Kermani, B.G., Schiffman, S.S. and Nagle, H.T. (2005) 'Performance of the Levenberg–Marquardt neural network training method in electronic noise applications', *Sensors and Actuators B*, 110, pp. 13–22.
- Kewalramani, M.A. and Gupta, R. (2006) 'Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks', *Automation in Construction*, 15(3), pp. 374–379.
- Khan, A.I., Bandopadhyaya, T.K. and Sharma, S. (2008) 'Genetic algorithm based backpropagation neural network performs better than backpropagation neural network in stock rates prediction', *International Journal of Computer Science and Network Security (IJCSNS)*, 8(7), pp. 162–166.
- Khan, M.I. (2012) 'Predicting properties of High Performance Concrete containing composite cementitious materials using Artificial Neural Networks', *Automation in Construction*, 22, pp. 516–524.
- Kim, G.H., Yoon, J.E., An, S.H., Cho, H.H. and Kang, K.I. (2004) 'Neural network model incorporating a genetic algorithm in estimating construction costs', *Building and Environment*, 39(11), pp. 1333–1340.
- Kim, J., Kim, D.K., Feng, M.Q. and Yazdani, F. (2004) 'Application of neural networks for estimation of concrete strength', *Journal of Materials in Civil Engineering*, 16(3), pp. 257–264.
- Kisi, O. (2008) 'Constructing neural network sediment estimation model using a data-driven algorithm', *Mathematics and Computers in Simulation*, 79(1), pp. 94–103.
- Kitano, H. (1990) 'Empirical Studies on speed of convergence of Neural Network training using Genetic Algorithms', *Proceedings of the 8th National Conference on Artificial Intelligence(AAAI-90)*, July 29–August 3, 1990, Boston, Massachusetts, pp. 789–795.
- Kostic, S. and Vasovic, D. (2015) 'Prediction model for compressive strength of basic concrete mixture using artificial neural networks', *Neural Computing and Applications*, 26(5), pp.1005–1024.
- Lagaros, N.D., Papadrakakis, M. and Kokossalakis, G. (2002) 'Structural optimization using evolutionary algorithms', *Computers and Structures*, 80(7–8), pp. 571–589.
- Lai, S. and Serra, M. (1997) 'Concrete strength prediction by means of neural network', *Construction and Building Materials*, 11(2), pp. 93–98.

- Lee, J., Lee, J. and Cho, B. (2012) 'Effective prediction of thermal conductivity of concrete using neural network method', *International Journal of Concrete Structures and Materials*, 6(3), pp. 177–186.
- Lee, S. (2003) 'Prediction of concrete strength using artificial neural networks', *Engineering Structures*, 25(7), pp. 849–857.
- Legates, D.R. and Davis, R.E. (1997) 'The continuing search for an anthropogenic climate change signal: Limitations of correlation-based approaches', *Geophysical Research Letters*, 24(18), pp. 2319–2322.
- Legates, D.R. and McCabe, G.J. (1999) 'Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation', *Water Resources Research*, 35(1), pp. 233–241.
- Li, G., Alnuweiri, H., Wu, Y. and Li, H. (1993) 'Acceleration of back propagation through initial weight pre-training with delta rule', *IEEE International Conference on Neural Networks*. San Francisco CA. 28<sup>th</sup> Mar-1<sup>st</sup> April 1993. IEEE, pp. 580–585.
- Li, Z. (2011) *Advanced Concrete Technology*. New Jersey: John Wiley & Sons, Inc.
- Lima, C., Sastry, K., Goldberg, D.E. and Lobo, F. (2005) 'Combining competent crossover and mutation operators: A probabilistic model building approach', *Proceedings of the 7th annual conference on Genetic and evolutionary computation (GECCO'05)*. Washington DC, USA. 25<sup>th</sup>-29<sup>th</sup> June 2005. New York: ACM, pp. 735–742.
- Lin, W.-Y., Lee, W.-Y. and Hong, T.-P. (2003) 'Adapting crossover and mutation rates in genetic algorithms', *Journal of Information Science and Engineering*, 19, pp. 889–903.
- Liu, W. and Chung, C. (2014) 'Enhancing the prediction accuracy of the water stage using a physical-based model and an artificial neural network-genetic algorithm in a river stage', *Water*, 6, pp. 1642–1661.
- Maier, H.R. and Dandy, G.C. (2000) 'Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications', *Environmental Modelling and Software*, 15(1), pp. 101–124.
- Maier, H.R., Jain, A., Dandy, G.C. and Sudheer, K.P. (2010) 'Methods used for the development of neural networks for the prediction of water resources variables in river systems: Current status and future directions', *Environmental Modelling & Software*, 25(8), pp. 891–909.
- Majdi, A. and Beiki, M. (2010) 'Evolving neural network using a genetic algorithm for predicting the deformation modulus of rock masses', *International Journal of Rock Mechanics & Mining Sciences*, 47, pp. 246–253.
- Marar, K. and Eren, O. (2011) 'Effect of cement content and water/cement ratio on fresh concrete properties without admixtures', *International Journal of Physical Sciences*, 6(24), pp. 5752–5765.

- Martinez-Martinez, V., Gomez-Gil, F.J., Gomez-Gil, J. and Ruiz-Gonzalez, R. (2015) 'An Artificial Neural Network based expert system fitted with Genetic Algorithms for detecting the status of several rotary components in agro-industrial machines using a single vibration signal', *Expert Systems with Applications*, 42(17–18), pp. 6433–6441.
- Masters, T. (1993) *Practical Neural Network Recipes in C++*. New York: Academic Press Inc.
- MATLAB 7.1 'Matrix Laboratory (R14 SP3)', The MathsWorks, Inc.
- Mavrovouniotis, M. and Yang, S. (2015) 'Training neural networks with ant colony optimization algorithms for pattern classification', *Soft Computing*, 19(6), pp. 1511–1522.
- McCall, J. (2005). 'Genetic algorithms for modeling and optimization', *Journal of Computational and Applied Mathematics*, 184(1), pp. 205–222.
- Mellit, A., Kalogirou, S. A. and Drif, M. (2010) 'Application of neural networks and genetic algorithms for sizing of photovoltaic systems', *Renewable Energy*, 35(12), pp. 2881–2893.
- Mermerdas, K. and Arbili, M.M. (2015) 'Explicit formulation of drying and autogenous shrinkage of concretes with binary and ternary blends of silica fume and fly ash', *Construction and Building Materials*, 94, pp.371–379.
- Miao, X., Chu, J., Zhang, L. and Qiao, J. (2013) 'An evolutionary neural network approach to simple prediction of dam deformation', *Journal of Information & Computational Science*, 10(5), pp. 1315–1324.
- Mindess, S., Toung, J. F., and Darwin, D. (2003) *Concrete*, 2nd ed., Upper Saddle River, NJ; USA: Pearson Education.
- Molga, E.J. (2003) 'Neural network approach to support modeling of chemical reactors: problems, resolutions, criteria of application', *Chemical Engineering and Processing: Process Intensification*, 42(8–9), pp. 675–695.
- Momeni, E., Nazir, R., Jahed Armaghani, D. and Maizir, H. (2014) 'Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN', *Measurement*, 57, pp. 122–131.
- Montana, D.J. and Davis, L. (1989) 'Training feedforward neural networks using genetic algorithms', In *IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence*, 1, pp 762-767.
- Montgomery, D.C. (2009) *Design and Analysis of Experiments*. New Delhi: Wiley India (P) Ltd.
- Moriassi, D.N., Arnold, J.G, Van Liew, M.W., Bingner, R.L., Harmel, R.D. and Veith, T.L. (2007) 'Model evaluation guidelines for systematic quantification of accuracy in watershed simulations', *Transactions of American Society of Agricultural and Biological Engineers (ASABE)*, 50(3), pp. 885–900.

- Muhit, I.B. (2013) ‘Dosage limit determination of superplasticizing admixture and effect evaluation on properties of concrete’, *International Journal of Scientific & Engineering Research*, 4(3), pp. 1–4.
- Myers, R.H., Montgomery, D.C. and Anderson-Cook, C.M. (2009) *Response Surface Methodology: Process and Product Optimization using Designed Experiment*. 3<sup>rd</sup> Ed. New Jersey: A John Wiley & Sons, Inc., Publication.
- Naderpour, H., Kheyroddin, A. and Ghodrati Amiri, G. (2010) ‘Prediction of FRP-confined compressive strength of concrete using artificial neural networks’, *Composite Structures*, 92(12), pp. 2817–2829.
- Najafi, Z. and Ahangari, K. (2013) ‘The prediction of concrete temperature during curing using regression and artificial neural network’, *Journal of Engineering*, 2013, Article ID: 946829, 5 pages.
- Najigivi, A., Khaloo, A., Zad, A.I. and Rashid, S.A. (2013) ‘An artificial neural networks model for predicting permeability of nano silica-rice husk ash ternary blended concrete’, *International Journal of Concrete Structures and Materials*, 7(3), pp. 225–238.
- Nash, J.E. and Sutcliffe, J.V. (1970) ‘River flow forecasting through conceptual models part I- A discussion of principles’, *Journal of Hydrology*, 10(3), pp. 282–290.
- Nasseri, M., Asghari, K. and Abedini, M.J. (2008) ‘Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network’, *Expert Systems with Applications*, 35(3), pp. 1415–1421.
- Nawari, N.O., Liang, R. and Nusairat, J. (1999) ‘Artificial intelligence techniques for the design and analysis of deep foundations’, *Electronic Journal of Geotechnical Engineering*, 4.
- Nawi, N.M., Atomi, W.H. and Rehman, M.Z. (2013) ‘The Effect of Data Pre-Processing on Optimized Training of Artificial Neural Networks’, *Procedia Technology*, 8, pp. 33–40.
- Nguyen, L.B., Nguyen, A.V., Ling, S.H. and Nguyen, H.T. (2013) ‘Combining genetic algorithm and Levenberg-Marquardt algorithm in training neural network for hypoglycemia detection using EEG signals’, *Proceedings of the 35<sup>th</sup> Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Osaka, Japan. 3<sup>rd</sup> Jul – 7<sup>th</sup> Jul 2013. IEEE, pp.5386–5389.
- Ni, H.G. and Wang, J.Z. (2000) ‘Prediction of compressive strength of concrete by neural networks’, *Cement and Concrete Research*, 30(8), pp. 1245–1250.
- Ni, J., Zhang, C. and Liu, M. (2010) ‘The Application of Neural Network Optimized by Genetic Algorithm in Water Quality Prediction’, *Proceedings of the Second International Conference on Information Science and Engineering (ICISE)*. Hangzhou, China. 4<sup>th</sup>-6<sup>th</sup> December 2010. IEEE, pp. 1582–1585.

- Nikoo, M., Moghadam, F.T. and Sadowski, L. (2015) 'Prediction of Concrete Compressive strength by Evolutionary Artificial Neural Networks', *Advances in Material Science and Engineering*, Volume 2015, pages 8.
- Nikoo, M., Zarfam, P. and Nikoo, M. (2012) 'Determining the displacement in concrete reinforcement building using evolutionary artificial neural networks', *World Applied Sciences Journal*, 16(2), pp. 1699–1708.
- Noori, R., Abdoli, M.A., Farokhnia, A. and Abbasi, M. (2009) 'Results uncertainty of solid waste generation forecasting by hybrid of wavelet transform-ANFIS and wavelet transform-neural network', *Expert Systems with Applications*, 36(6), pp. 9991–9999.
- Noori, R., Hoshyaripour, G., Ashrafi, K. and Nadjar-Araabi, B. (2010) 'Uncertainty analysis of developed ANN and ANFIS models in prediction of carbon monoxide daily concentration', *Atmospheric Environment*, 44(4), pp. 476–482.
- Olden, J.D. and Jackson, D.A. 2002 'Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks', *Ecological Modelling*, 154(1-2), pp. 135–150.
- Olden, J.D., Joy, M.K. and Death, R.G. (2004) 'An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data', *Ecological Modelling*, 178(3–4), pp. 389–397.
- Ozkan, C. and Erbek, F.S. (2003) 'The comparison of activation functions for Multispectral Landsat TM Image Classification', *Photogrammetric Engineering and Remote Sensing*, 69(11), pp. 1225–1234.
- Oztas, O., Pala, M., Ozbay, E., Kanca, E., Caglar, N. and Bhatti, M.A. (2006) 'Predicting the compressive strength and slump of high strength concrete using neural network', *Construction and Building Materials*, 20(9), pp. 769–775.
- Paliwal, M. and Kumar, U.A. (2011) 'Assessing the contribution of variables in feed forward neural network', *Applied Soft Computing*, 11(4), pp. 3690–3696.
- Pandurangan, K. and Kothandaram, S. (2012) 'Effect of coarse aggregate size and shape on the strength and flow characteristics of Self-compacting Concrete', *ICI Journal*, pp. 1–7.
- Parichatprecha R. and Nimityongskul, P. (2009) 'Analysis of durability of high performance concrete using artificial neural networks', *Construction and Building Materials*, 23(2), pp. 910–917.
- Park, W. (2013) 'Genetic-algorithm-based mix proportion design method for recycled aggregate concrete', *Transactions of Canadian Society for Mechanical Engineering*, 37(3), pp. 345–354.
- Patuwo, E., Hu, M.Y. and Hung, M.S. (1993) 'Two group classification problem using neural networks', *Decision Sciences*, 24(4), pp.825–846.
- Pencheva, T., Atanassov, K. and Shannon, A. (2009). 'Modelling of a Stochastic Universal Sampling Selection Operator in Genetic Algorithms Using Generalized



- Nets', *Proceedings of the 10<sup>th</sup> International Workshop on Generalized Nets*. Sofia, Bulgaria. 5<sup>th</sup> December 2009. pp. 1–7.
- Pendharkar, P.C. and Rodger, J.A. (2003) 'Technical efficiency based selection of learning cases to improve the forecasting efficiency of neural networks under monotonicity assumption', *Decision Support Systems*, 36(1), pp. 117–136.
- Peyghami, M.R. and Kanduzi, R. (2012) 'Predictability and forecasting automotive price based on a hybrid train algorithm of MLP neural network', *Neural Computing and Applications*, 21(1), pp. 125–132.
- Pie, Y. and Xia, Y. (2012) 'Design of Reinforced Cantilever Retaining Walls using Heuristic Optimization Algorithms', *Procedia Earth and Planetary Science*, 5, pp. 32–36.
- Plawiak, P. and Mariarz, W. (2014) 'Classification of tea specimens using novel hybrid artificial intelligence methods', *Sensors and Actuators B: Chemical*, 192, pp. 117–125.
- Plawiak, P. and Tadeusiewicz, R. (2014) 'Approximation of phenol concentration using novel hybrid computational intelligence methods', *International Journal of Applied Mathematics and Computer Science*, 24(1), pp. 165–181.
- Pofale, A.D. and Quadri, S. R. (2013) 'Effective utilization of crusher dust in concrete using Portland Pozzolana Cement', *International Journal of Scientific and Research Publications*, 3(8), pp. 1–9.
- Poulton, M. M. (2002) 'Neural networks as an intelligence amplification tool: a review of applications', *Geophysics*, 67(3), pp. 979–993.
- Priddy, K. L. and Keller, P. E. (2005) *Artificial Neural Networks: An Introduction*. Bellingham, Washington, USA: SPIE-The International Society for Optical Engineering.
- Ray, C. and Klindworth, K. (2000) 'Neural networks for agrichemical vulnerability assessment of rural private wells', *Journal of Hydrologic Engineering*, 5(2), pp. 162–171.
- Razali, N. M. and Geraghty, J. (2011) 'Genetic algorithm performance with different selection strategies in solving TSP', *Proceedings of the World Congress on Engineering*. London, U.K. 6<sup>th</sup> -8<sup>th</sup> July 2011. pp. 1134–1139.
- Rocha, I., Parente, Jr., E., Melo, A.M.C. (2014) 'A hybrid shared/distributed memory parallel genetic algorithm for optimization of laminate composites', *Composite Structures*, 107, pp. 288–297.
- Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). *Learning internal representations by error propagation*. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Volume 1: Foundations, MIT Press, Cambridge, MA.

- Sabet, F.A., Libre, N.A. and Shekarchi, M. (2013) 'Mechanical and durability properties of self consolidating high performance concrete incorporating natural zeolite, silica fume and fly ash', *Construction and Building Materials*, 44, pp. 175–184.
- Saemi, M., Ahmadi, M. and Varjani, A.Y. (2007). 'Design of neural networks using genetic algorithm for the permeability estimation of the reservoir', *Journal of Petroleum Science and Engineering*, 59(1-2), pp. 97–105.
- Sahoo, B. and Maity, D. (2007) 'Damage assessment of structures using hybrid neuro-genetic algorithm', *Applied Soft Computing*, 7(1), pp. 89–104.
- Samani, N., Gohari-Moghadam, M. and Safavi, A.A. (2007) 'A simple neural network model for the determination of aquifer parameters', *Journal of Hydrology*, 340 (1–2), pp. 1-11.
- Sbartai, Z.M., Laurens, S., Viriyametanont, K., Balayssac, J.P. and Arliguie, G. (2009) 'Non-destructive evaluation of concrete physical condition using radar and artificial neural networks', *Construction and Building Materials*, 23(2), pp.837–845.
- Sedki, A., Ouazar, D., and El Mazoudi, E. (2009) 'Evolving neural network using real coded genetic algorithm for daily rainfall-runoff forecasting', *Expert Systems with Applications*, 36(3), pp. 4523–4527.
- Senouci, A.B. and Al-Ansari, M.S. (2009). 'Cost optimization of composite beams using genetic algorithms', *Advances in Engineering Software*, 40(11), pp. 1112–1118.
- Sexton, R.S., Dorsey, R.E. and Johnson, J.D. (1999) 'Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing', *European Journal of Operational Research*, 14, pp. 589–601.
- Shamseldin, A. Y., Nasr, A. E. and O'Connor, K. M. O. (2002) 'Comparison of different forms of multi-layer feed-forward neural network method used for river flow forecasting', *Hydrology and Earth System Sciences*, 6(4), pp. 671–684.
- Shibata, K. and Ikeda, Y. (2009) 'Effect of number of hidden neurons on learning in large-scale layered neural networks', *Proceedings of the ICROS-SICE International Joint Conference 2009 (ICCASSICE '09)*. Fukuoka, Japan. 18<sup>th</sup>-21<sup>st</sup> August 2009. IEEE, pp. 5008–5013.
- Siddique, R., Aggarwal, P. and Aggarwal, Y. (2011) 'Prediction of compressive strength of self-compacting concrete using bottom ash using artificial neural networks', *Advances in Engineering Software*, 42(10), pp. 780–786.
- Sietsma, J. and Dow, J. (1991). Creating artificial neural networks that generalize. *Neural Networks*, 4 (1), pp. 67–79.
- Sobhani, J., Najimi, M., Pourkhorshidi, A.R. and Parhizkar, T. (2010) 'Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models', *Construction and Building Materials*, 24(5), pp.709–718.

- Sola, J. and Sevilla, J. (1997) 'Importance of input data normalization for the application of neural networks to complex industrial problems', *IEEE Transactions on Nuclear Science*, 44(3), pp. 1464–1468.
- Sovil, D., Kvanicka, V. and Pospichal, J. (1997) 'Introduction to multi-layer feed-forward neural networks', *Chemo metrics and Intelligent Laboratory Systems*, 39(1), pp. 43–62.
- Srinivasulu, S. and Jain, A. (2006) 'A comparative analysis of training methods for artificial neural network rainfall-runoff models', *Applied Soft Computing*, 6(3), pp. 295–306.
- Staufer, P. and Fisher, M.M. (1997) *Spectral pattern recognition by a two-layer perceptron: effects of training set size*. In: Kanellopoulos, I., Wilkinson, G.G., Roli, F. and Austin, J., editors. *Neuro-computation in remote sensing data analysis*. London: Springer; 1997, pp. 105–16.
- Subasi, S. (2009) 'Prediction of mechanical properties of cement containing class C fly ash by using artificial neural network and regression technique', *Scientific Research and Essay*, 4(4), pp. 289–297.
- Sudarsana Rao, H. and Ramesh Babu, B. (2006) 'Optimized column design using genetic algorithm based neural networks', *Indian Journal of Engineering and Material Sciences*, 13, pp. 503–511.
- Sudarsana Rao, H. and Ramesh Babu, B. (2007) 'Hybrid neural network model for the design of beam subjected to bending and shear', *Sadhana*, 32(5), pp. 577–586.
- Sudrasana Rao, H., Ghorpade, Vaishali G. and Mukherjee, A. (2006) 'A genetic algorithm based back propagation network for simulation of stress-strain response of ceramic-matrix-composites', *Computers and Structures*, 85(5-6), pp.330–339.
- Suryadi, A., Triwulan and Aji, P. (2011) 'Predicting the setting time of self compacting concrete using artificial neural networks (Anns) with the various of learning rate coefficient', *Journal of Applied Sciences Research*, 7(3), pp. 314–320.
- Sutton, R. (1986) 'Two problems with backpropagation and other steepest-descent learning procedures for networks', *Proceedings of the 8<sup>th</sup> Annual Conference Cognitive Science Society*, pp. 823–831.
- Tamura, S. and Tateishi, M. (1997) 'Capabilities of a four-layered feedforward neural network: four layers versus three', *IEEE Transactions on Neural Networks*, 8(2), pp. 251–255.
- Tan, M., He, G., Nie, F., Zhang, L. and Hu, L. (2014) 'Optimization of ultrafiltration membrane fabrication using backpropagation neural network and genetic algorithm', *Journal of the Taiwan Institute of Chemical Engineers*, 45(1), pp. 68–75.
- Tayfur, G., Erdem, T.K. and Kirca, O. (2014) 'Strength prediction of High-Strength Concrete by Fuzzy Logic and Artificial Neural Networks', *Journal of Materials in Civil Engineering*, 26(11).

- Tokar, A.S. and Johnson, P.A. (1999) 'Rainfall-runoff modeling using artificial neural networks', *Journal of Hydrologic Engineering*, 4(3), pp. 232–239.
- Topcu, I.B. and Saridemir, M. (2007) 'Prediction of properties of waste AAC aggregate concrete using artificial neural network', *Computational Materials Science*, 41(1), pp. 117–125.
- Uysal, M. and Tanyildizi, H. (2011) 'Predicting the core compressive strength of self compacting concrete (SCC) mixtures with mineral additives using artificial neural network', *Construction and Building Materials*, 25(11), pp. 4105–4111.
- Wang, S., Dong, X. and Sun, R. (2010) 'Predicting saturates of sour vacuum gas using artificial neural networks and genetic algorithms', *Expert Systems with Applications*, 37(7), pp. 4768–4771.
- Wang, Z., Fang, S. and Fu, S. (2012) 'ANN Synthesis Models Trained with Modified GA-LM Algorithm for ACPWs with Conductor Backing and Substrate Overlaying', *ETRI Journal*, 34(5), pp. 696–705.
- Whitley, D., Starkweather, T., and Bogart, C. (1990) 'Genetic algorithms and neural networks: optimizing connections and connectivity', *Parallel Computing*, 14(3), pp. 347–361.
- Wilamowski, B.M., Chen, Y. and Malinowski, A. (1999) 'Efficient algorithm for training neural networks with one hidden layer', *Proceedings of the International Joint Conference on Neural Networks (IJCNN'99)*. Washington DC, USA. 10<sup>th</sup>-16<sup>th</sup> July 1999. IEEE, pp. 1725–1728.
- Willmott, C.J. and Matsuura, K. (2005) 'Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance', *Climate Research*, 30, pp. 79–82.
- Wu, K.L. and Chau, K.W. (2006) 'A flood forecasting neural network model with genetic algorithm', *International Journal of Environment and Pollution*, 28(3-4), pp. 261–273.
- Xue, Y., Cheng, L., Mou, J. and Zhao, W. (2014) 'A new fracture prediction method by combining genetic algorithm with neural network in low-permeability reservoirs', *Journal of Petroleum Science and Engineering*, 121, pp. 159–166.
- Yegnanarayana, B. (2001). *Artificial neural networks*. New Delhi: Prentice-Hall of India Private Limited.
- Yeh, I.-C. (2007) 'Modeling slump flow of concrete using second-order regressions and artificial neural networks', *Cement & Concrete Composites*, 29(6), pp. 474–480.
- Yeh, I.-C. (1998) 'Modeling of strength of high-performance concrete using artificial neural networks', *Cement and Concrete Research*, 28(12), pp. 1797–1808.
- Yeh, I.-C. (2006) 'Exploring concrete slump model using artificial neural networks', *Journal of Computing in Civil Engineering*, 20(3), pp. 217–220.

- Yeh, I-C. (2008) 'Modeling slump of concrete with fly ash and superplasticizer', *Computers and Concrete*, 5(6), pp.559–572.
- Yinghua, W. and Chang, X. (2010) 'Using Genetic Artificial Neural Network to Model Dam Monitoring Data', *Proceedings of the Second International Conference on Computer Modeling and Simulation*. Sanya, Hainan, China. 22<sup>nd</sup>-24<sup>th</sup> January 2010. IEEE, pp. 3–7.
- Yu, J.-B., Yu, Y., Wang, L.-N., Yuan, Z. and Ji, X. (2016) 'The knowledge modeling system of ready-mixed concrete enterprise and artificial intelligence with ANN-GA for manufacturing production', *Journal of Intelligent Manufacturing*, 27(4), pp. 905–914.
- Yuan, Z., Wang, L.N. and Ji, X. (2014) 'Prediction of concrete compressive strength: Research on hybrid models genetic based algorithms and ANFIS', *Advances in Engineering Software*, 67, pp. 156–163.
- Zadeh, L.A. (1994). 'Fuzzy logic, neural networks, and soft computing', *Communications of the ACM*, 37(3), pp. 77–84.
- Zavrtanik, N., Prosen, J., Tusar, M. and Turk, G. (2016) 'The use of artificial neural networks for modeling air void content in aggregate mixture', *Automation in Construction*, 63, pp.155–161.
- Zhang, G., Patuwo, B.E. and Hu, M.Y. (1998) 'Forecasting with artificial neural networks: The state of the art', *International Journal of Forecasting*, 14(1), pp. 35–62.
- Zhang, Q. and Wang, C. (2008) 'Using genetic algorithm to optimize artificial neural networks: A case study on Earthquake prediction', *Proceedings of the IEEE Second International Conference on Genetic and Evolutionary Computing*. Hubei, China. 25<sup>th</sup>-26<sup>th</sup> September 2008. IEEE, pp.128–131.
- Zheng, L. (1999) 'Prediction and classification with neural network models', *Sociological Methods and Research*, 27(4), pp. 499–524.

## **Appendix I**

**Design mix proportions and slump data collected from first RMC batching plant (493 datasets)**

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
1	150	0	860	638	522	2.50	150	90
2	155	0	858	640	521	2.45	145	90
3	380	0	725	720	370	0.00	190	120
4	200	0	820	620	540	2.20	170	125
5	250	0	614	764	516	1.30	150	140
6	150	70	806	654	558	2.50	164	140
7	100	120	840	620	534	2.00	160	140
8	150	70	806	650	562	2.50	164	140
9	300	0	843	697	343	2.50	178	140
10	300	0	824	702	346	2.40	188	140
11	330	0	800	500	582	3.50	172	150
12	330	0	806	490	580	3.30	165	150
13	120	120	840	428	682	3.00	160	150
14	380	0	810	500	500	3.50	180	150
15	330	0	806	486	580	3.50	166	150
16	380	0	790	550	490	3.80	160	150
17	330	0	806	490	580	3.50	160	150
18	380	0	788	550	490	4.00	170	150
19	330	0	800	500	582	3.50	160	150
20	120	120	840	438	682	3.00	160	150
21	300	0	722	604	588	3.40	166	150
22	330	0	806	490	580	3.30	156	150
23	330	0	800	500	582	3.90	170	150
24	380	0	810	500	490	3.80	174	150
25	330	0	810	582	500	3.70	165	150
26	330	0	810	582	500	3.70	175	150
27	330	0	810	500	582	3.80	170	150
28	330	0	806	490	580	3.60	160	150
29	330	0	800	582	500	3.70	158	150
30	330	0	800	500	582	3.50	170	150
31	380	0	790	550	500	4.00	170	150
32	330	0	806	490	580	3.50	158	150
33	330	0	806	490	580	3.40	160	150
34	380	0	810	500	490	4.00	170	150
35	140	120	780	516	526	1.50	186	150
36	300	0	722	604	588	3.20	160	150
37	380	0	790	550	490	3.50	180	150
38	330	0	806	490	580	3.50	162	150
39	330	0	810	500	582	3.70	165	150
40	300	0	722	604	588	3.20	155	150
41	380	0	810	500	500	4.00	176	150
42	380	0	810	600	500	4.00	175	150
43	380	0	790	550	490	3.80	168	150
44	120	120	850	438	680	2.50	170	150

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
45	380	0	790	550	490	4.00	180	150
46	300	0	700	762	468	3.50	150	150
47	350	0	800	500	582	3.70	166	150
48	300	0	722	604	588	3.50	162	150
49	380	0	790	550	490	4.00	170	150
50	300	0	722	604	588	3.20	166	150
51	380	0	810	500	490	4.00	176	150
52	330	0	806	580	510	3.30	160	150
53	330	0	800	500	582	3.00	170	150
54	120	120	850	438	680	3.00	150	150
55	120	120	800	536	536	2.50	166	150
56	300	0	800	600	600	3.80	160	150
57	120	120	860	438	600	2.00	170	150
58	380	0	790	550	490	4.00	160	150
59	380	0	750	550	490	4.00	170	150
60	330	0	800	500	582	3.60	168	150
61	120	120	850	438	680	2.50	175	150
62	330	0	800	500	582	3.50	166	150
63	120	120	860	438	680	2.00	180	150
64	340	0	800	500	582	3.80	170	150
65	380	0	790	550	490	4.00	178	150
66	330	0	806	490	580	3.50	156	150
67	120	120	860	536	536	2.60	180	150
68	120	120	850	438	680	2.50	150	150
69	330	0	750	528	582	4.20	170	150
70	330	0	806	490	580	3.30	160	150
71	330	0	800	500	582	3.30	170	150
72	120	120	840	438	680	2.00	176	150
73	380	0	810	500	500	4.10	180	150
74	330	0	800	500	582	3.80	170	150
75	380	0	810	500	500	4.00	180	155
76	200	60	770	604	560	2.30	170	155
77	250	100	776	530	558	2.90	163	155
78	240	0	850	590	500	2.40	160	155
79	330	0	800	580	500	4.00	176	155
80	330	0	800	592	510	3.80	170	155
81	250	100	776	530	558	3.00	162	155
82	120	180	850	412	618	2.60	174	160
83	100	200	835	515	530	2.50	175	160
84	220	110	774	586	530	4.00	170	160
85	220	110	756	590	545	4.00	164	160
86	240	100	790	570	508	3.80	160	160
87	330	0	736	602	552	3.90	170	160
88	250	100	712	602	558	3.80	166	160
89	330	0	716	598	552	3.70	170	160
90	250	110	756	562	562	3.70	168	160



S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
91	224	100	779	568	568	3.90	168	160
92	224	100	740	594	552	4.00	170	160
93	220	110	800	580	510	3.80	165	160
94	400	0	718	730	429	4.00	152	160
95	220	110	742	576	552	3.90	170	160
96	330	0	716	600	554	4.20	174	160
97	300	0	786	580	540	3.50	168	160
98	244	106	786	590	500	3.90	158	160
99	330	0	755	600	540	3.80	163	160
100	244	100	796	600	508	3.50	170	160
101	180	120	810	600	476	3.00	179	160
102	400	0	662	598	556	4.50	176	160
103	300	0	800	600	476	3.90	168	160
104	250	100	776	530	558	3.00	162	160
105	220	110	790	600	476	3.80	160	160
106	220	110	790	580	510	3.50	156	160
107	210	113	746	580	530	4.50	164	160
108	248	100	726	584	544	4.20	170	160
109	300	0	800	570	536	3.70	170	160
110	224	100	760	570	570	3.70	165	160
111	350	0	716	598	552	4.20	174	160
112	224	100	752	590	550	3.60	166	160
113	220	110	790	600	476	4.20	162	160
114	220	110	800	580	504	3.60	156	160
115	250	100	726	592	558	4.00	170	160
116	240	100	796	580	500	3.80	156	160
117	224	100	742	600	560	3.30	166	160
118	220	110	800	600	480	3.60	170	160
119	340	30	676	598	552	4.20	170	160
120	250	100	716	602	558	4.00	160	160
121	350	0	810	500	490	4.30	180	160
122	224	100	774	568	568	3.56	172	160
123	240	100	790	570	506	3.80	160	160
124	220	110	780	600	486	4.20	174	160
125	424	0	546	730	486	5.20	170	160
126	220	110	780	600	486	3.20	166	160
127	330	0	736	602	550	4.00	170	160
128	220	110	780	600	486	4.00	174	160
129	330	0	756	600	540	3.80	170	160
130	220	110	780	600	486	3.80	174	160
131	330	0	736	602	550	3.50	170	160
132	320	0	750	600	558	4.00	174	160
133	224	100	742	600	560	3.80	160	160
134	220	110	782	526	586	3.40	170	160
135	220	110	782	550	576	2.90	158	160
136	180	120	752	604	560	2.50	164	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
137	240	100	790	570	510	4.20	165	160
138	270	90	746	580	530	4.50	164	160
139	320	40	700	604	558	2.50	136	160
140	224	100	742	594	560	3.50	162	160
141	296	30	796	572	520	3.80	158	160
142	248	100	726	584	544	4.20	176	160
143	224	100	750	600	540	3.60	176	160
144	224	100	741	600	560	3.20	150	160
145	220	110	755	550	560	4.20	170	160
146	250	100	730	572	572	4.00	170	160
147	330	0	746	592	552	3.60	160	160
148	240	110	786	584	498	3.90	158	160
149	220	110	774	586	530	3.80	170	160
150	220	110	786	580	540	3.50	178	160
151	224	100	740	594	552	4.20	170	160
152	270	100	790	586	490	4.40	170	160
153	246	100	744	620	520	4.20	176	160
154	224	100	774	568	568	3.70	164	160
155	244	100	790	570	508	3.80	160	160
156	180	120	762	600	460	2.80	170	160
157	240	110	796	580	490	3.80	158	160
158	220	110	786	580	540	3.50	168	160
159	220	110	790	580	510	3.40	156	160
160	224	100	742	594	560	3.80	162	160
161	350	0	742	580	550	3.90	170	160
162	240	100	796	580	500	3.80	160	160
163	224	100	742	596	550	3.90	166	160
164	224	100	746	598	552	4.00	174	160
165	350	0	716	600	554	4.20	176	160
166	330	0	746	592	552	3.60	170	160
167	270	110	750	540	520	4.20	170	160
168	320	0	736	584	544	4.10	170	160
169	240	110	760	570	520	3.50	166	160
170	250	106	750	556	556	3.80	170	160
171	224	100	742	600	560	3.50	162	160
172	270	80	716	598	552	2.00	172	160
173	250	100	730	580	558	4.40	170	160
174	244	100	790	570	508	4.00	168	160
175	250	100	730	606	540	3.90	168	160
176	426	0	790	590	552	5.20	168	160
177	240	100	796	580	500	3.80	158	160
178	350	0	716	596	552	3.30	170	160
179	250	100	716	602	558	4.00	166	160
180	244	106	780	570	510	3.70	160	160
181	250	100	780	580	510	3.80	160	160
182	246	104	744	580	536	4.00	165	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
183	330	0	790	600	540	3.60	156	160
184	180	120	762	600	460	3.00	182	160
185	330	0	728	600	442	3.80	162	160
186	150	120	840	620	500	2.00	160	160
187	220	110	780	600	488	3.80	169	160
188	250	100	780	586	510	4.20	160	160
189	300	0	810	610	494	3.30	126	160
190	244	106	780	510	510	3.80	160	160
191	300	0	800	600	480	3.60	160	160
192	224	100	742	606	562	4.00	164	160
193	120	100	780	638	590	3.00	152	160
194	220	110	762	564	560	3.00	164	160
195	426	0	642	590	548	5.00	168	160
196	240	100	800	580	504	3.60	156	160
197	330	20	776	530	558	3.10	170	160
198	220	110	800	580	510	3.50	156	160
199	244	100	790	570	508	4.00	165	160
200	224	100	742	600	560	3.80	166	160
201	250	100	716	600	558	3.80	166	160
202	350	30	676	598	558	4.20	170	160
203	240	100	796	580	500	3.70	160	160
204	250	105	750	555	545	4.00	160	160
205	240	100	790	580	510	4.20	170	160
206	320	30	710	594	552	3.70	166	160
207	330	0	740	600	556	3.80	174	160
208	250	100	786	580	500	3.80	160	160
209	224	100	752	610	550	4.00	176	160
210	224	100	742	596	550	3.80	166	160
211	224	100	740	594	552	4.00	172	160
212	224	100	752	610	544	4.40	176	160
213	180	120	762	578	576	3.00	160	160
214	296	30	770	580	530	4.30	175	160
215	450	0	642	576	540	5.80	180	160
216	300	0	810	610	494	3.60	142	160
217	320	50	710	594	554	4.20	174	160
218	300	0	800	600	510	3.40	156	160
219	224	106	750	570	570	3.80	166	160
220	240	100	796	580	500	3.70	156	160
221	220	110	742	606	562	3.80	170	160
222	280	100	692	594	550	4.00	176	160
223	410	0	700	594	506	4.00	180	160
224	224	100	742	596	560	3.80	166	160
225	220	110	800	570	544	4.00	164	160
226	424	0	546	730	486	4.50	166	160
227	120	160	860	526	526	2.60	190	160
228	426	0	642	590	552	5.00	170	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
229	240	100	786	590	500	3.80	158	160
230	250	100	760	550	558	4.00	164	160
231	270	50	752	590	550	3.90	164	160
232	320	0	800	550	540	4.40	164	160
233	330	0	716	598	552	4.20	172	160
234	224	100	752	580	560	3.90	162	160
235	370	0	746	580	530	4.80	170	160
236	244	100	790	570	520	3.80	178	160
237	250	100	730	606	540	4.40	178	160
238	220	110	776	570	544	4.00	174	160
239	250	100	762	576	520	4.20	174	160
240	250	100	760	550	550	4.00	172	160
241	250	100	764	580	536	4.20	174	160
242	270	110	750	582	498	4.00	166	160
243	250	105	760	545	545	3.80	160	160
244	220	100	800	580	504	3.70	165	160
245	240	100	790	580	500	3.40	160	160
246	220	110	782	526	586	3.70	170	160
247	300	0	800	600	500	3.80	160	160
248	220	110	780	600	486	3.80	178	160
249	270	100	750	582	498	4.00	166	160
250	220	110	790	590	476	3.80	164	160
251	180	130	755	552	548	3.90	170	160
252	248	100	726	584	544	4.10	170	160
253	250	100	750	560	558	4.10	170	160
254	250	100	746	572	572	3.92	170	160
255	220	110	800	580	510	3.50	160	160
256	220	110	774	586	530	4.00	160	160
257	220	110	794	592	504	3.80	170	160
258	230	100	740	594	552	4.00	168	160
259	330	0	604	604	544	3.90	164	160
260	240	110	786	580	500	4.00	160	160
261	220	110	800	600	480	3.40	152	160
262	220	100	810	600	486	3.60	150	160
263	250	100	730	606	540	3.80	166	160
264	300	0	840	580	500	3.80	160	160
265	130	200	900	500	520	2.50	175	160
266	244	100	768	596	508	3.80	156	160
267	220	110	780	600	510	3.40	156	160
268	220	110	742	586	542	4.00	175	160
269	250	100	730	572	572	4.00	160	160
270	250	100	776	550	538	3.00	163	160
271	224	100	760	570	570	3.80	164	160
272	270	100	706	586	540	4.40	176	160
273	350	0	748	576	572	3.40	165	160
274	250	100	736	582	558	4.10	164	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
275	380	0	662	598	556	4.20	180	160
276	244	100	790	570	520	3.50	160	160
277	270	100	750	560	530	4.00	165	160
278	220	110	800	580	504	3.40	160	160
279	300	0	780	550	560	3.80	175	160
280	246	100	744	620	520	4.00	170	160
281	224	100	752	576	576	3.90	156	160
282	224	100	760	570	570	3.80	165	160
283	340	40	676	608	552	3.00	164	160
284	244	100	780	590	500	3.80	160	160
285	220	110	800	570	544	3.40	156	160
286	224	100	742	600	560	3.50	160	160
287	350	0	716	598	552	4.00	170	160
288	330	0	810	700	400	3.96	160	160
289	450	0	642	576	540	5.80	186	160
290	380	90	590	594	550	5.50	176	160
291	300	0	780	654	516	3.50	156	160
292	220	115	755	550	555	4.00	164	160
293	320	30	700	604	558	2.40	150	160
294	244	100	764	580	536	4.00	178	160
295	220	110	790	600	476	3.70	170	160
296	300	0	815	590	510	3.50	156	160
297	220	110	774	586	530	3.20	160	160
298	250	100	726	592	558	4.00	174	160
299	244	106	780	570	510	3.80	160	160
300	224	100	742	594	560	3.60	156	160
301	220	110	772	530	570	3.80	172	160
302	250	100	758	560	540	3.80	174	160
303	220	100	752	590	550	3.60	166	160
304	220	110	780	600	486	4.00	175	160
305	220	110	782	526	586	4.00	170	160
306	418	0	546	730	486	4.50	166	160
307	220	110	800	590	514	3.50	156	160
308	250	100	762	576	520	4.20	170	160
309	300	0	800	600	490	3.83	179	160
310	330	0	786	548	548	4.20	170	160
311	220	110	742	576	552	4.10	170	160
312	300	0	774	568	568	3.70	170	160
313	280	100	780	580	494	3.70	160	160
314	250	100	760	550	550	4.00	166	160
315	224	100	742	606	562	3.80	160	160
316	380	0	790	550	590	4.00	179	160
317	220	110	800	586	500	3.50	160	160
318	240	100	790	580	500	3.80	160	160
319	220	110	790	590	476	3.80	160	160
320	270	80	716	600	558	4.30	172	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
321	300	0	790	572	572	3.50	150	160
322	330	0	716	600	558	3.80	166	160
323	250	100	726	592	558	4.00	176	160
324	350	0	700	580	552	3.50	170	160
325	270	100	694	594	550	4.10	170	160
326	250	100	800	600	476	3.80	162	160
327	330	0	770	560	560	3.80	167	160
328	250	100	730	572	572	4.10	170	160
329	320	0	780	580	530	3.80	164	160
330	426	0	642	590	552	5.00	168	160
331	220	110	800	580	504	3.50	160	160
332	307	0	740	740	450	2.46	177	165
333	272	68	800	550	510	3.50	165	170
334	272	68	800	550	510	3.50	164	170
335	330	80	626	752	436	4.50	170	170
336	450	0	672	632	420	5.20	180	170
337	450	0	672	632	428	5.30	170	170
338	450	0	652	612	408	5.50	180	170
339	250	0	550	754	520	1.50	180	170
340	272	68	800	550	510	3.80	175	170
341	450	0	672	632	428	5.00	170	170
342	450	0	472	632	428	5.50	170	170
343	418	0	800	434	556	4.60	174	180
344	240	80	806	714	393	3.20	159	180
345	430	0	830	510	430	4.50	160	190
346	210	113	750	710	385	3.55	105	190
347	250	0	614	764	515	1.30	150	140
348	380	0	810	500	500	4.00	180	150
349	180	0	870	624	530	2.20	166	115
350	330	0	800	500	582	3.70	170	150
351	120	120	840	438	680	3.00	180	150
352	300	0	722	604	588	3.40	156	150
353	380	0	810	500	500	4.00	178	150
354	330	0	810	582	500	3.70	170	150
355	380	0	790	550	490	3.80	170	150
356	330	0	806	490	580	3.50	170	150
357	330	0	806	490	580	3.30	170	150
358	380	0	810	500	490	4.00	180	150
359	120	120	840	438	680	3.00	160	150
360	320	0	790	600	610	3.40	160	155
361	250	100	726	592	558	4.10	174	160
362	220	110	790	596	476	3.80	160	160
363	418	0	546	730	588	4.18	166	160
364	224	100	742	600	560	3.30	150	160
365	210	90	746	580	530	4.50	164	160
366	200	100	810	590	514	3.60	150	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
367	220	100	742	606	562	3.80	170	160
368	300	0	800	570	536	3.80	150	160
369	220	110	810	570	508	3.50	160	160
370	224	100	750	600	540	3.80	176	160
371	220	110	756	590	544	4.00	170	160
372	180	120	752	604	560	3.00	180	160
373	250	100	730	606	540	3.80	174	160
374	244	106	780	580	500	3.80	160	160
375	280	0	814	568	568	3.56	170	160
376	250	100	726	592	558	3.90	166	160
377	230	100	800	500	582	3.70	170	160
378	224	100	752	590	550	3.90	156	160
379	330	0	756	600	540	3.80	174	160
380	250	100	726	592	558	4.10	170	160
381	400	0	662	598	556	4.80	176	160
382	220	100	810	600	486	3.50	152	160
383	240	110	780	556	520	3.10	166	160
384	250	0	762	576	520	4.20	170	160
385	224	100	740	594	552	4.20	174	160
386	220	110	770	610	486	3.80	175	160
387	290	30	742	600	560	3.80	170	160
388	250	100	790	576	500	3.80	164	160
389	220	100	790	600	476	3.80	164	160
390	220	110	780	600	488	3.20	169	160
391	250	100	716	598	552	4.20	168	160
392	220	110	786	586	540	3.50	168	160
393	120	160	860	526	526	2.60	175	160
394	200	100	790	530	564	2.50	180	160
395	250	110	750	545	545	4.00	165	160
396	224	100	774	568	568	3.56	170	160
397	250	100	716	598	552	4.20	172	160
398	220	110	780	600	510	3.50	156	160
399	300	0	760	590	540	3.50	170	160
400	270	100	706	586	540	4.00	173	160
401	360	0	736	602	552	3.00	166	160
402	250	100	746	572	572	4.40	170	160
403	380	0	662	598	556	4.30	186	160
404	410	0	700	594	508	4.00	180	160
405	320	0	726	584	544	4.10	170	160
406	330	0	736	602	552	3.50	172	160
407	246	104	764	580	536	4.20	174	160
408	220	110	742	596	532	4.10	174	160
409	250	100	762	576	520	4.20	178	160
410	272	68	800	570	544	3.40	156	160
411	270	110	730	570	510	4.30	175	160
412	240	100	796	590	500	4.00	158	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
413	350	0	716	596	552	3.50	170	160
414	300	0	800	600	476	3.91	168	160
415	272	68	800	550	510	3.80	176	170
416	450	0	672	632	428	5.20	180	170
417	450	0	672	632	428	5.50	170	170
418	272	68	800	550	510	3.80	174	170
419	450	0	672	632	428	5.50	174	170
420	430	0	838	510	430	4.50	160	190
421	180	0	860	640	522	1.90	165	115
422	200	0	850	640	535	1.80	170	120
423	100	120	840	620	534	2.20	160	140
424	330	0	806	486	580	3.50	170	150
425	330	0	800	500	582	3.20	166	150
426	300	0	722	604	588	3.50	170	150
427	330	0	810	490	578	3.50	156	150
428	330	0	749	530	581	4.20	170	150
429	380	0	810	500	490	4.30	180	150
430	300	0	722	604	588	3.20	156	150
431	330	0	800	500	582	3.60	164	150
432	330	0	806	489	581	3.50	160	150
433	140	120	780	526	516	1.50	150	150
434	220	110	782	526	586	4.00	170	155
435	330	0	806	500	570	3.50	160	155
436	120	110	775	640	593	3.00	152	155
437	330	0	806	490	580	3.30	165	155
438	200	60	770	610	552	2.30	170	155
439	270	90	745	580	530	4.00	170	160
440	285	95	738	495	530	4.56	185	160
441	270	60	790	600	476	4.10	160	160
442	350	0	710	610	554	2.00	154	160
443	320	0	725	583	544	4.10	170	160
444	220	110	801	601	478	3.60	170	160
445	220	110	802	579	508	3.70	165	160
446	330	0	785	550	550	4.20	170	160
447	224	100	742	600	560	3.20	150	160
448	220	120	737	525	543	4.08	180	160
449	180	120	830	588	500	3.20	150	160
450	330	20	775	530	559	3.10	170	160
451	300	50	786	590	490	3.70	158	160
452	300	0	802	600	480	3.60	160	160
453	220	110	795	593	544	3.80	170	160
454	350	0	820	524	524	4.00	166	160
455	420	0	710	584	480	5.00	176	160
456	240	110	796	580	490	3.60	158	160
457	220	110	800	580	504	3.60	160	160
458	220	110	780	598	515	3.50	156	160



S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP(kg/ m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
459	220	110	780	520	591	3.80	170	160
460	250	100	774	572	572	4.50	172	160
461	230	110	770	610	486	3.80	175	160
462	224	100	752	576	576	3.80	160	160
463	250	100	744	580	536	4.00	164	160
464	224	100	752	580	560	3.90	166	160
465	350	0	740	562	562	4.50	162	160
466	250	100	718	600	556	4.20	170	160
467	220	120	737	525	543	4.42	180	160
468	340	30	710	594	552	3.70	166	160
469	220	110	732	586	542	3.80	170	160
470	250	100	758	560	540	3.50	151	160
471	330	100	670	560	540	5.00	179	160
472	250	100	746	582	548	3.00	170	160
473	424	0	546	730	486	5.20	180	160
474	220	110	772	530	570	3.80	176	160
475	300	0	810	550	550	3.80	166	160
476	250	100	715	603	558	4.00	166	160
477	220	110	742	576	552	4.00	180	160
478	220	115	755	550	560	3.90	170	160
479	240	110	760	570	520	3.00	166	160
480	220	100	810	600	486	3.60	152	160
481	220	110	800	580	504	3.60	164	160
482	220	110	811	571	505	3.50	156	160
483	320	0	670	660	540	3.80	160	160
484	320	30	710	594	558	3.70	166	160
485	250	100	745	580	535	4.00	164	160
486	220	110	810	570	508	3.50	156	160
487	300	0	700	730	468	3.30	150	165
488	272	68	800	509	551	3.80	174	165
489	300	0	790	550	520	3.57	160	165
490	246	120	740	690	430	3.50	164	165
491	450	0	672	632	428	5.50	182	170
492	450	0	672	632	428	5.40	170	170
493	450	0	652	612	408	5.50	170	170

## **Appendix II**

**Design mix proportions and slump data collected from second RMC batching plant (100 datasets)**

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP (kg/m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
1	120	120	800	624	530	2.60	170	150
2	120	120	810	438	600	3.00	162	150
3	120	120	800	650	530	2.50	170	160
4	120	100	806	654	552	2.50	160	130
5	120	120	800	680	438	2.70	176	155
6	130	200	820	580	520	2.50	175	160
7	150	0	860	638	522	2.50	165	100
8	150	120	842	620	498	2.00	160	155
9	160	120	811	501	538	3.64	181	160
10	170	100	762	616	570	3.00	175	160
11	180	120	819	600	471	2.90	165	160
12	180	120	765	580	570	3.00	160	160
13	180	120	828	600	500	3.20	150	160
14	180	130	790	545	560	2.90	180	160
15	186	64	900	674	348	1.70	189	160
16	210	120	780	495	578	3.70	166	155
17	215	120	737	525	543	3.85	180	160
18	215	120	737	525	543	3.35	180	160
19	220	110	779	600	487	3.20	166	160
20	220	110	779	600	487	3.80	166	160
21	220	100	774	568	568	3.56	170	160
22	220	100	750	605	535	3.80	166	160
23	220	100	730	596	576	3.70	166	160
24	220	110	800	589	545	3.50	160	160
25	224	100	742	600	560	3.20	162	160
26	224	100	742	600	560	4.00	170	160
27	224	100	794	558	558	3.70	162	160
28	226	50	782	618	496	2.50	170	150
29	230	60	820	475	600	3.00	166	150
30	230	60	830	600	480	3.00	164	150
31	240	100	795	581	500	3.70	156	160
32	240	100	790	500	580	3.40	160	155
33	240	0	850	590	500	2.30	140	150
34	240	100	796	580	500	3.80	156	165
35	240	0	850	590	500	2.30	140	155
36	240	0	849	590	501	2.60	165	155
37	250	100	765	580	536	4.20	174	160
38	250	100	760	578	520	4.20	170	160
39	250	100	726	590	560	4.10	174	160
40	250	100	715	599	558	3.80	166	160
41	250	100	715	599	552	4.20	168	160
42	250	100	712	604	555	3.80	166	160
43	250	110	799	600	477	4.00	170	160
44	250	100	762	556	556	3.00	150	160

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP (kg/m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
45	250	100	736	520	560	4.00	166	160
46	250	120	750	545	545	3.60	160	155
47	255	120	724	515	533	4.13	180	150
48	270	90	760	536	547	3.00	166	160
49	270	100	789	587	490	4.10	170	160
50	270	100	790	586	490	4.10	170	160
51	270	80	715	600	559	4.30	172	160
52	272	68	800	509	551	3.80	174	170
53	272	68	798	553	511	3.80	176	170
54	272	68	800	550	570	3.50	165	160
55	272	68	800	550	570	3.50	165	170
56	272	68	800	508	552	4.00	174	160
57	280	100	592	594	550	4.00	176	160
58	300	0	721	605	588	3.20	160	150
59	300	0	801	599	475	3.90	168	160
60	300	0	811	610	495	3.70	160	160
61	300	0	722	602	588	3.20	166	150
62	300	0	702	760	467	3.30	150	155
63	300	0	810	600	480	4.08	166	160
64	300	0	810	610	494	3.70	160	160
65	320	30	711	603	557	3.70	168	160
66	320	30	710	604	558	3.70	170	160
67	330	0	729	600	440	3.80	162	160
68	330	0	806	489	581	3.30	160	150
69	330	0	734	604	550	4.00	170	160
70	330	0	716	597	553	3.70	170	160
71	330	100	800	500	582	3.70	170	150
72	330	0	810	490	580	3.50	170	150
73	330	100	798	500	584	3.70	170	150
74	330	0	805	492	580	3.30	156	150
75	330	0	806	489	581	3.40	160	150
76	330	0	800	510	593	3.80	170	155
77	330	0	800	500	582	3.50	160	155
78	330	100	650	599	531	5.50	180	155
79	330	0	798	500	584	3.50	164	150
80	330	100	652	564	564	4.50	170	160
81	330	0	810	500	582	3.70	165	155
82	340	30	675	599	552	4.20	170	155
83	350	0	716	594	552	3.50	170	160
84	350	0	716	597	552	3.50	170	160
85	350	0	753	575	545	3.40	170	155
86	350	0	746	576	542	3.40	157	160
87	350	0	740	562	562	4.50	162	155
88	350	30	675	600	550	4.20	170	155
89	350	30	676	598	552	4.20	170	150
90	350	0	710	608	556	2.00	154	160

---

---

S.No	Cement (kg/m <sup>3</sup> )	PFA (kg/m <sup>3</sup> )	Sand (kg/m <sup>3</sup> )	CA(20mm) (kg/m <sup>3</sup> )	CA(10mm) (kg/m <sup>3</sup> )	SP (kg/m <sup>3</sup> )	Water (kg/m <sup>3</sup> )	Slump (mm)
91	350	0	810	500	490	4.30	180	150
92	350	0	722	604	588	3.50	162	160
93	350	0	710	610	554	2.00	154	155
94	360	0	736	602	552	4.00	142	160
95	380	0	810	500	490	4.20	170	150
96	380	0	754	580	496	3.80	166	150
97	380	0	754	580	496	4.00	166	160
98	380	0	810	500	490	4.30	182	150
99	450	0	672	632	426	5.50	174	170
100	450	0	652	612	408	5.50	174	170

---

## **Publication List**

## Papers published based on this work

### SCI Journals

1. Chandwani, V., Agrawal, V., and Nagar, R. (2015) 'Modeling slump of ready mix concrete using genetic algorithms assisted training of Artificial Neural Networks', *Expert Systems with Applications*, 42(2), pp. 885–893.

### Non SCI Journals

1. Chandwani, V., Agrawal, V., and Nagar, R. (2014) 'Modeling Slump of Ready Mix Concrete using genetically evolved Artificial Neural Networks', *Advances in Artificial Neural Systems*, Volume 2014, Article ID 629137, 9 pages, <http://dx.doi.org/10.1155/2014/629137>.
2. Chandwani, V., Agrawal, V., and Nagar, R. (2013) 'Applications of Soft Computing in Civil Engineering: A Review', *International Journal of Computer Applications (IJCA)*, 81(10), pp. 13-20.

### International Conferences

1. Agrawal V., Chandwani V., Nagar R. and Singh S. (2014) 'Hybrid Artificial Neural Networks Aided Conceptual Design of Industrial Roof Trusses', *Proceedings of the Fifth International Joint Conference on Advances in Engineering and Technology-AET 2014*, December 27, 2014, Kochi, India.

## **Author's Bio-data**



### **Author's Bio-data**

The author obtained his Bachelor's degree in Civil Engineering from Malaviya Regional Engineering College, Jaipur (Presently Malaviya National Institute of Technology Jaipur) in the year 1996. He was awarded the University Gold Medal for standing first in the order of merit in civil engineering. He enrolled for M.Tech degree in Structural Engineering in the year 2006 and completed his M.Tech degree in the year 2009 from Malaviya National Institute of Technology Jaipur. The author joined as Assistant Engineer in Water Resources Department, Government of Rajasthan in the year 1998 and is presently holding the post of Executive Engineer. The author was actively engaged in the design of irrigation structures like dams, anicuts, canals etc., during his posting at the Investigation Design and Research (ID&R) unit of Water Resources Department.