**Artificial Neural Network Modeling, Results and Discussion**

**6.1 Introduction**

Artificial Neural Networks are similar in structure and functioning to the Biological Neural Networks (BNNs). The basic computing elements that perform data processing in Artificial Neural Network are the neurons. Each neuron in the network receives input from many other neurons. It changes its internal state (or activation) based on the present input. Every neuron sends its output to many other neurons. The neurons learn from their experimental data to detain the linear and non linear trends in the complex data, so that they provide reliable predictions for even new situations containing partial information. The processing ability of the network is stored in the interneuron connection strengths called weights. The weights are acquired through an adaptation process. The usage of ANN models to predict the highly desirable compressive strength of SIFCON has proved to be more advantageous over the conventional methods for providing a solution to inverse problems that are non linear in nature. Multilayer feed forward back propagation neural networks have been in use for predicting the strength characteristics of concrete, slump of ready mixed concrete and high strength concrete (Dias and Pooliyadda, 2001). In this chapter, a new feed forward and back propagation ANN design using three types of optimisation techniques namely Levenberg-Marquardt, Steepest descent, Genetic and Polynomial curve fitting model with R2≈1 are proposed for accurate prediction of strength characteristics of SIFCON concrete such as compressive, split tensile and flexural strengths having different fibre fractions and different curing periods.

The structure of this chapter is as follows. In section 6.2 ANN architecture and its training process have been presented. The proposed SDANN, LMANN, GANN models and Curve Fitting model are presented in section 6.3, 6.4, 6.5 and 6.6 respectively. Section 6.6 also includes the Goodness of fit statistics and confidence bounds on the coefficients. The results of post analysis are discussed in section 6.7 and finally, the conclusions have been summarized in section 6.8.

**6.2 ANNs Architecture and Training process**

It has been concluded from the existing literature that it is not possible for a single neuron to learn the complete behavior of model using all the training sets. Therefore, a suitable ANN model is required to be designed by utilizing the interconnection of neurons in order to perform the desired task. Two types of ANNs are used for training the experimental data and they are:

1. Feed forward ANN, also called Multilayer Perceptron (MLP) and
2. Recurrent neural networks.

Feed forward networks have been used to process data in this research work. The term, feed forward relates to the way neural network processes and recalls the patterns. In this network, neurons are only forward connected i.e., each layer of the NN contains connections only to the next layer. Feed forward ANNs have significant computational powers because the interconnections have no closed paths or loops, whereas connections between neurons form a directed cycle in case of recurrent ANNs. For feed forward neural networks, the most widely used learning algorithm uses the back propagation which is a form of supervised training. In feed forward networks, during the training phase the network must be provided with sample inputs and expected outputs. The expected outputs are compared against the actual outputs for a given input. From the anticipated data, the Back propagation training algorithm calculates the error and adjusts the weights of the various layers backwards from output layer to the input layer. A three layer feed forward ANN has been shown in Fig.6. 1a.

The feed forward equation corresponding to the inputs (xp) and outputs (yk) of a three-layer feed forward Multilayer perceptron model (Fig.6.1a) are described by Eq.6.1 in matrix form.

wji

ukj

bias =1

x0

bias =1

h0

Input layer

(x1,..xi..xP )

Hidden layer

(h1..hj..hM )

Output layer

(y1,y2..yk..yK)

**.**

**:**

**:**

x × w

connection weight

bias connection weight

yk= (x;w)

Fig.6. 1 Three layer feed forward network

The kth  output of ANN for an input vector (x) and weight matrix (W) is given by

 (6.1)

The neuron output at kth node in matrix form at layer, L is given by

 (6.2)

The neuron output at jh node and at hidden layer, L-1 is given by

 (6.3)

Eq. 6.3 can be written in matrix form as

 (6.4)

The input vector x with p number of variables and bias (x0 =1) is represented by

 (6.5)

Where,

σ(.) **=** activation function.

wjo = bias in the input- hidden nodes connection

wko =bias in the hidden-output nodes connection

h(.) = sigmoid activation function

p = Number of real input nodes

M = Number of real hidden nodes

K = Number of output nodes

WM= Matrix of weights wji between input- hidden nodes

u= Matrix of weights ukj between hidden-output nodes

yk = Output vector at kth node



Activation function of a node defines the output of that particular node of an input or set of inputs given to this node. The activation function used often is the sigmoid function, a continuous version of the sign function. It’s an S shaped curve that can take any real valued number and map it into a value between 0 and 1, but never exactly at those limits.

**ANN training process:**

The ANN is the system which contains the number of artificial neurons connected in a network with certain set of weights. The neuron is the processing element which receives the inputs and processes them for singular value function (activation function). The two main aspects in the ANN are to collect the data by using the learning process and to store that collected data by using the interneuron connection strength. The ANN is trained offline using the Steepest Descent (SD), Conjugate Gradient (CG), Genetic Algorithm (GA), and Levenberg–Marquardt (LM) back propagation algorithms which is nonlinear optimization algorithm involving the second-order derivatives. It requires more memory for the processing, because it functions as the square of the number of weights for error calculation. It mostly works with the summed error functions widely used for estimation (regression) applications. The performance is measured by mean-squared error (MSE) as shown in Fig. 6.2. The Mean Squared Error (MSE) is the average squared difference between outputs and targets. Lower values of MSE are better and zero MSE means no error.

The connections between elements (neurons) largely determine the network function and the network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Figure 6.2 illustrates such a situation. Here, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network. Neural networks can also be trained to solve problems that are difficult to be solved by conventional computers or human beings.

Target data obtained from experimental study

Output from ANN model

ANN including connections (referred to as weights)

between neurons

Input to the proposed ANN model

Adjust weights

using optimization techniques (LM,SD,GA,Conjugate gradient etc.,)

Fig.6.2 Weights adjustment procedure in the proposed ANN Model

The training procedures developed for different optimization algorithms (SD, CG, GA, LM, etc.) are given in the following sections. The derivatives of objective functions and Jacobian matrices of the various optimization techniques are given in chapter 5. The final equations of those objective functions (or cost functions) have been used in the development of ANN models for prediction of SIFCON strength characteristics.

**6.3 Development of SD based ANN for prediction of SIFCON strength**

In this optimization algorithm training instances are shown to the model one at a time. The model makes a prediction for a training instance, the error is then calculated and the model is updated in order to reduce the error for the next prediction. This procedure is used to find the set of such coefficients in a model that result in the smallest error by using the training data. In each iteration, the coefficients called weights (w) in machine learning language are updated using the following equation:

weight being optimized (w) = previous epoch weight – (learning rate) × error for the model (6.6)

The implementation of artificial neural network training using Steepest Descent **(**SD) algorithm involves two steps and these are: i) Calculation of Jacobian matrix; ii) training process design.

1. **Calculation of Jacobian matrix**

SD is the process of minimizing a function by following the gradients of the cost function. This requires a prior knowledge about the form of the cost function as well as its derivative so that from a given point the gradient is known. This helps to move in that particular direction e.g. towards the minimum value.

In structural engineering, the objective function to be minimized is of the form

 (6.7)

where, rj is the residual due to ANN model output.

In structural engineering to fit the actual data, the function (Eq. 6.7) is used to measure discrepancy between the computed and measured quantities. Each residual represents the difference between computed quantities and their measured counter-part. The residual vector,  can be written as

**** (6.8)

Using this notation, the objective function given in Eq. 6.7 can be written as



The Jacobian matrix,is a representation of the derivatives of the objective function which will be m × n matrix. In this case  is n-dimensional vector and the Jacobian matrix is given by

 (6.9)

The gradient (g) of the objective function can be written in terms of the Jacobian as  (6.10)

1. **SD based Training process design**

The SD based training process is implemented using the following steps:

**Step 1:** Estimate the error with the initial weights that are randomly generated

**Step 2:** Do an update using Eq. 6.7 to adjust the weights.

**Step 3:** With the new weights, estimate the mean squared error (MSE).

**Step 4:** Update the following Equation of Steepest Descent algorithm to adjust weights



 (6.11)

Where, is the learning constant (step size) and W is the vector consisting of w1, w2.....wn.

**Step 5:** With the new weights evaluate the mean squared error

**Step 6:** If the current MSE has increased as a result of the update, then retract the step such as reset the weight vector to the previous value and decrease combination coefficient,  by a factor of 10 or by the same factor as step 4.

**Step 7:** Go to step 2 with new weights until the current MSE is smaller than the required value.

##### **6.4 Development of Levenberg-Marquardt algorithm based ANN training process**

The implementation of artificial neural network training using the Levenberg*-*Marquardtalgorithm involves two steps: i) Calculation of observation or Jacobian matrix and ii) ANN training process design.

The Jacobian matrix derivation is not given in this section but the final equation is provided at the appropriate place of the algorithm. The important steps followed in the Levenberg*-*Marquardt algorithm based ANN training process are given in this section. The basic idea of Levenberg*-*Marquardt algorithm is that it performs a combined training process around the area with complex curvature. The Levenberg*-*Marquardt algorithm switches to the Steepest Descent algorithm, until the local curvature is proper to make a quadratic approximation; then it approximately becomes the Gauss-Newton algorithm, which can speed up the convergence significantly. The training process using Levenberg-Marquardt algorithm is designed using the following steps:

**Step 1**: a) Initialize the weights for the network.

              b) Apply the input vector to the input layer of the designed ANN.

             c) Output is generated by propagating the data forward to the network and calculate the error between the target output and actual output.

             d) Evaluate the error using Eq. 6.12 with the initial weights that are randomly generated.

 (6.12)

Where,e(w) defined as a sum of the squared errors over all the output units k for *all* the training examples *n* (e.g. input vector, x and weight matrix ,w)

xn = Number of real input nodes.

M = Number of real hidden nodes.

y(xn ,w) = ANN model Output vector

dn= ANN model desired or target value

rj are called residuals and is given by rj = target value - ANN output

Each residual (rj) represents the difference between computed quantities and their measured counterpart, and therefore represent the function of sought parameters.

**Step 2**: Compute the Jacobian matrix (J) from the first-order partial derivatives using Eq.6.9.

**Step 3**: Compute the weight update of the network as   directed by Eq. 6.13 to adjust weights.

 (6.13)

where,

represents the update equation at time, t+1

W is the vector consisting of w1, w2.....wn

 is a residual vector =****

 is combination coefficient and is always positive

*I* is Identity matrix

is the error in the estimation of W and is given by

 (6.14)

While implementing the above equation, Levenberg-Marquardt switches between the two algorithms (Gauss Newton and Steepest Descent) as follows.

When  or near to zero,

Equation (6.13) becomes the Gauss Newton algorithm and is given by



When 

Eq. 6.13 becomes the Steepest Descent algorithm and is given by



Where, is the learning constant (or step size) and is given by



**Step 3**: With the new weights, compute the error using Eq. 6.12.

**Step 4**: If the new mean squared error is increased as a result of the update, then retract the step (such as reset the weight vector to the previous value) and increase combination coefficient, µ by a factor of 10 or by the same factor as ‘step 2’.

**Step 5**: If the new estimated mean squared error is decreased as a result of the update, then accept the step (such as keep the latest weight vector as the fresh value) and decrease combination coefficient, µ by a factor of 10 or by the same factor as ‘step 4’.

**Step 6**: Go to step ‘2’ with the new weights until the current total error is smaller than the required value.

* 1. Development of GA implementation for ANN training process

GA an optimization method associated with high probability to reach a global minimum solution (Goldberg D.E., 1989). The property of ANN to learn, when combined with the property of Genetic Algorithm to converge to a global optimum, results to more competent algorithms. The method of implementing ANN with GA is discussed below.

After deciding upon the number of input, output nodes and hidden layer/nodes, the weight updating is undertaken with GA. Initially Xt is defined as aset of weights in the designed neural network at time instant t and these weights form the chromosomes of parent in the selected population. With the initialized population of weights, the output of the neural network is evaluated and found the fitness of individual parent based on network output.

The parents from this initial population were selecting for mating pool based on a specific criteria and fitness value. The output of the mating pool resulted in new population with a set of updated weights, Xt+1, whose fitness is further evaluated and the above process is repeated till the desired fitness parent is generated. The parents obtained from required fitness were the solution or optimal trained weights for the designed network. Cross over (introducing a random weight as chromosome to a parent) operation was performed in order to converge towards global optimum, and hence the rate of crossover has to be very low. The stepwise procedure for implementation of GAANN for prediction of strength characteristics of SIFCON are given below and graphical representation of the proposed GANN implementation is shown in Fig. 6.3.

Step 1: Fix the size of Input, Output and Hidden layer nodes.

Step 2: Determine the number of network weights in the network and create initial population.

Step 3: The chromosome size of individual parent in the population should be made equal to size of network weights.

Step 4: Initialize the chromosomes of the parents (i.e. initialize the weights) randomly.

Step 5: Evaluate the network output with every parent (i.e. a set of weights) in the generated population.

Step 6: Determine the fitness of parents based on fitness function or objective function given in Eq. 6.12

Step 7: Based on the predefined criteria select parents for mating pool and based on the requirement crossover has to be done.

Step 8: The generated children from the mating pool will now become new set of population for next generation.

Step 9: Step ‘5’ is repeated if required fitness parent is not generated and when the required fitness is achieved the iterations will be stopped.

Step 10: The final parent chromosomes which hold weight information are optimal weights for the network considered.

Set the values of population size and chromosome size

Initialize the chromosomes (i.e. weights of the ANN)

Evaluate the fitness of parents (i.e. set of weights) within the population using a cost function

Fix the size of input/hidden/output layer nodes

Parent with required fitness is obtained

Chromosomes of the fitted parent are optimal weights

Cross over of parents based on a specific criteria

Mutation required

Yes

Mutation based on a criterion

No

Yes

No

Fig. 6.3 GAAN implementation procedure

**6.6 Development of Curve Fitting Model for prediction of SIFCON strength**

Curve fitting method is the process of building a curve, or mathematical function, that has the best fit to a series of discrete data points to obtain estimates for other data points. The main application of curve fitting is predicting the values of dependent variable (or output), and may also include extrapolation beyond input data points or interpolation between the input data points. There are two general approaches for curve fitting. They are: i) Least Squares regression and ii) Interpolation. When the measured data exhibits a significant degree of scatter, least squares regression derives a single curve that represents the general trend of the data. When the datais very precise, the strategy in interpolation is to pass a curve or a series of curves through each of the points.

**Polynomial Regression:**

In all engineering applications, the measured data is not possibly well represented by a straight line due to non linear problem. For these cases a curve is better suited to fit the data. The least squares method can readily be extended to fit (or model) the data to higher order polynomials. For example, a 2nd order polynomial (quadratic) is considered to fit the data and is given by



The residuals (ei) between the model and the data are given by :



A residual is the distance of a point from the modelled curve. The sum of squares of the residual (e2) or Sum of the Squared Error (SSE) is given by

 (6.15)

The objective is to minimize the SSE between the measured ‘ymeasure’ and the ‘ymodel’ calculated with the model.

To obtain the least squared error, the unknown coefficients p1, p2, and p3 must yield zero on computing the first order derivatives of SSE. Therefore, the coefiicients p1, p2 and p3 needs to be computed such that SSE is minimised. By taking the derivative for the equation SSE with respect to the three polynomial coefficients

 (6.16)

The above three linear equations with three unknowns (p1, p2 and p3), is a homogeneous set of equations and can be solved. Rearranging the above equations and writing them in matrix form



(6.17)

To fit this data to an mth order polynomial,



A system of (m+1) × (m+1) linear equations must be solved for determining the coefficients of the mth order polynomial using least squares approximation method.



(6.18)

**6.6.1 Goodness of fit statistics and confidence bounds on the coefficients**

The curve fitting results are evaluated using goodness of fit test to see how close the curve is matching with the data and also to estimate the certainty (95% confidence bounds**)** of the predicted value**.** Usually the goodness of fit is quantified with the following parameters:

1. Sum of Squared Error (SSE)
2. R- Square
3. Root Mean Squared Error (RMSE)

Sum of Squared Error (SSE) represents the sum of the squares of the vertical distances of the points from the curve. It is expressed in the unit as that of squared values of Y. R- Square (R2) quantifies goodness of fit. R-square is a fraction between 0.0 and 1.0, and has no units. Higher R2 values (≈ 1) indicate that the curve comes closer to the data. R2 is computed from the sum of the squares of the distances of the points from the best-fit curve determined by nonlinear regression. This sum-of-squared error value is called SSEreg, which is in the units of the Y- axis squared. To turn R2 into a fraction, the results are normalized to the sum of the square of the distances of the points from a horizontal line through the mean of all Y values. This value is called SSEtot. R2 is calculated using the following equation:

  (6.19)

Root Mean Squared Error (RMSE) represents the standard deviation of the residuals, where residuals are the distance of the points from the fitted line. The equation 6.20 calculates the RMSE from the sum of squared error and degrees of freedom

 (6.20)

where, Q is number of data points and M is number of parameters

**6.6.2 Steps followed in the implementation of Polynomial Curve Fitting method for prediction of strength characteristics of SIFCON**

The following are the steps required for the implementation of Polynomial Curve Fitting method for prediction of strength characteristics of SIFCON.

**Step 1**: Use non linear regression in the curve fitting tool box to fit data to a model that defines variable output (y) as a function of input, x.

**Step 2**: Prepare the data and enter into the curve fitting tool box program.

**Step 3**: Choose the model that defines variable, y as a function of x.

**Step 4:** Choose initial values, since non linear regression is an iterative procedure.

**Step 5**: Perform the curve fit and interpret the best fit parameter values.

**Step 6**: Compute the Goodness of Fit statistics for analysis of the results using Eq. 6.15, 6.19 and 6.20

**6.7 Proposed ANN Models design**

The artificial network design consists of input and output vectors and the intermediate hidden layers (e.g. j and k). Once the input and output vectors are established for the ANN design, then a suitable configuration is selected. The first step in ANNs design involves the determination of the networks architecture such as networks type, the size of hidden layers and nodes. It is followed by the learning stage in which the weights of the network are iteratively adjusted until certain precision of the output is achieved. The number of hidden layers directly affects the performance of the network. Therefore, many experimental investigations are conducted. The number of hidden layers and number of neurons in layer are determined by trial and error method to provide the optimal result. This trial and error method is done until a perfect network configuration that meets the minimum mean squared error and high correlation coefficient criteria is achieved.

The following designs of ANN model architectures are found to be the best suited architectures for prediction of SIFCON strength characteristics with different configurations.

**ANN model architecture for Compressive strength prediction:**

1. LMANN 2-2-15-1 architecture
2. SDANN 2-4-14-1 architecture
3. GAANN 2-4-15-1 architecture
4. Polynomial of degree 2 curve fitting model

**ANN model architecture for Split tensile strength prediction:**

1. LMANN 2-1-10-1 architecture
2. SDANN 2-2-14-1 architecture
3. GAANN 2-1-14-1 architecture
4. Polynomial of degree 3 curve fitting model

**ANN model architecture for Flexural strength prediction:**

1. LMANN 2-2-10-1 architecture
2. SDANN 2-2-14-1 architecture
3. GAANN 2-1-10-1 architecture
4. Polynomial of degree 3 curve fitting model

The above ANN architectures have learned the relationship for predicting the strength of Slurry Infiltrated Fibrous concrete in different training epochs and they are implemented as per the respective procedures given in sections 6.3 to 6.6, the results have been compared and presented in section 6.8. The following sections present the detailed architectures of the above models.

**6.7.1 LMANN design**

A total of 15 ANN architectures are implemented as per the procedures given in section 6.5 for the LM based weight updation optimization technique and the results are analyzed for the optimal ANN model that gives the low MSE value. The proposed Levenberg Marquardt back-propagation Artificial Neural Network (LMANN) (2-2-15-1 design) for compressive strength consists of two input vectors namely percent fibre fraction and curing period, two hidden layers with 15 neurons each and compressive strength as output of the network (Fig. 6.4a). Figure 6.4a illustrates the LMANN 2-2-15-1 architecture model for compressive strength. Similarly Fig. 6.4b and 6.4c represents the LMANN 2-1-10-1 and LMANN 2-2-10-1 architecture models for split tensile and flexural strength predictions respectively. Figure 6.5a, 6.5b and 6.5c presents the simplified Levenberg-Marquardt architectures (2-2-15-1, 2-1-10-1 and 2-2-10-1) with inner details for compressive, split tensile and flexural strengths respectively. The detailed ANN architecture design is given in Fig. 6.6 for compressive strength prediction. Results of this design are compared with the other ANN models (polynomial curve fitting method, SD, and Genetic models) and are discussed and presented in section 6.8.1.

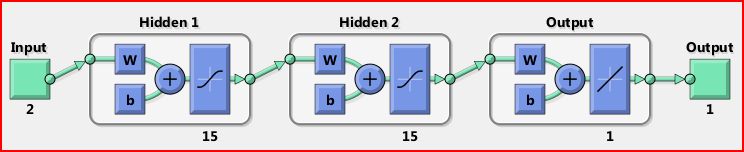


Fig. 6.4a LMANN 2-2-15-1 architecture model for compressive strength

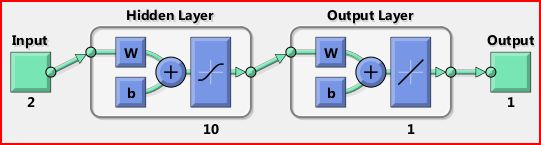


Fig. 6.4b LMANN 2-1-10-1 architecture model for split tensile strength

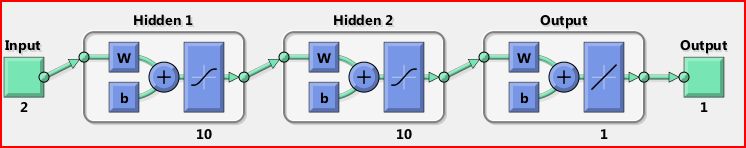


Fig. 6.4c LMANN 2-2-10-1 architecture model for flexural strength

Output layer

Compressive

strength

Bias

Bias

Days

% fibre

fraction

Two hidden layers with 15 nodes each

Input layer

15

14

2

1

15

2

1

14

Fig. 6.5a Simplified LM architecture 2-2-15-1 for compressive strength

Output

One hidden layer with 10 neurons

Input layer

days

% fibre fraction

Split tensile strength

Bias

1

2

9

10

Fig. 6.5b Simplified LM architecture 2-1-10-1 for split tensile strength

2 hidden layers each with 10 nodes

Input layer

Days

% fibre fraction

Bias

1

2

9

10

Output layer

Flexural Strength

1

2

9

10

Bias

2 Hidden Layers

Input layer

Days

% fibre fraction

Bias

1

2

9

10

Output

Flexural Strength

1

2

9

10

Bias

2 Hidden Layers

Input layer

Days

% fibre fraction

Bias

1

2

9

10

Output

Flexural Strength

1

2

9

10

Bias

Fig. 6.5c Simplified LM architecture 2-2-10-1 for flexural strength prediction

Input to neuron 2

**Curing period**

**Input to Neurons**

Input Neuron 1

**% fibre fraction**

**Output** of

Neuron

Hidden Neuron **j**

Hidden Neuron **k**

Bias = +1

Bias = +1

Bias = +1

Wji

Wlk

Wkj

Fig. 6.6 Detailed architecture of the proposed LMANN (2-2-15-1 design)

**6.7.2 SDANN model**

A total of 15 SDANN architectures were studied, and the best architectures (SDANN 2-4-14-1, SDANN 2-2-14-1 and SDANN 2-2-14-1) so chosen for SD algorithm are represented in Fig. 6.7a, Fig. 6.7b and 6.7c for prediction of compressive, split tensile and flexural strengths respectively. Figure 6.7a shows the 2-4-14-1 architecture, which contains 2 inputs, 4 hidden layers with 14 neurons in each layer and compressive strength as output of the network. SD algorithm was well trained with a total of 32 samples to predict the compressive strength. Figure 6.8a, Fig. 6. 8b and 6.8c illustrates the SDANN detailed architecture models for compressive, split tensile and flexural strengths respectively. Percentage fibre fraction and curing time represent the input vectors and compressive strength as the output in the architecture of Fig.6.8a.

The optimized configuration provides a lower RMS error. The training of the above networks was stopped early to avoid any over fitting effects and this was done as soon as the RMS error reached an area of no variation. A maximum of approximately 1000 iterations were found to be optimum in this work. However, it is important to note that the number of iterations needed for the ANN to learn the relation between the input/output mappings depends on the complexity of the problem. The RMS error is computed by using the following formula.

 (6.21)

where, DPY and OPY are the desired and the observed values, P is the number of training inputs and Y is the number of output nodes.

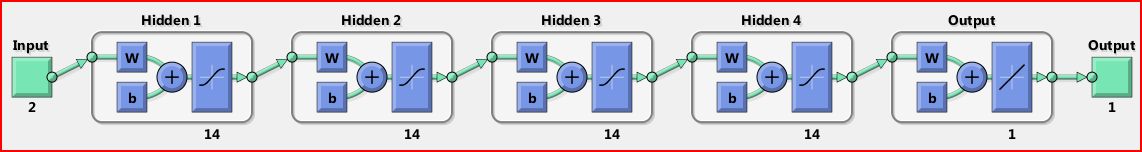


Fig. 6.7a SDANN 2-4-14-1 architecture for compressive strength

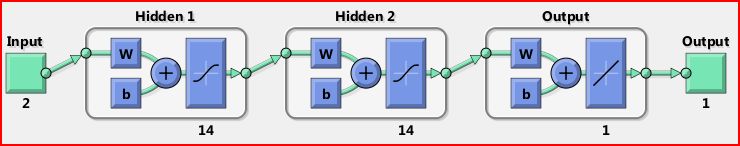


Fig. 6.7b SDANN 2-2-14-1 architecture for split tensile strength

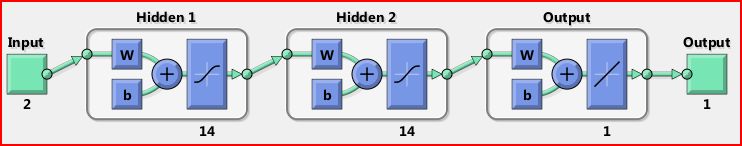


Fig. 6.7c SDANN 2-2-14-1 architecture for flexural strength

Output layer

Compressive strength

Bias

Bias

Bias

Bias

days

% fibre

fraction

14

13

2

1

14

2

1

14

2

1

14

2

1

13

13

13

Hidden layer

Input layer

Fig.6.8a Simplified SDANN architecture 2-4-14-1 for compressive strength

Hidden Layer

Input layer

Days

% fibre fraction

Bias

Output

Split tensile Strength

Bias

Fig. 6.8b Simplified SDANN architecture 2-2-14-1 for split tensile strength

Output

Hidden Layer

Input layer

Days

% fibre fraction

Flexural strength

Bias

1

2

1

2

3

1

Fig. 6.8c Simplified SDANN architecture 2-2-14-1 for flexural strength

**6.7.3 GAANN model**

The network construction was defined in terms of input and output vectors and the intermediate hidden layers. Once the input and output vectors were decided for the ANN design, then a suitable formation was selected. In selection of hidden layers number and nodes in the layer are selected by trial and error method. This trial and error method was continued until a perfect network configuration is obtained for prediction of strength characteristics of SIFCON. Figure 6.9a, Fig.6.9b and 6.9c illustrates the ANN architecture designs (GAANN 2-4-15-1, GAANN 2-1-14-1 and GAANN 2-1-10-1) for prediction of compressive strength, split tensile strength and flexural strengths respectively.

It was observed that the network with 15 neurons in each of 4 hidden layers (j, k, l and m) performed satisfactorily incase of compressive strength and accordingly a configuration of (GAANN 2-4-15-1) has been selected for this network model. The proposed GAANN 1-4-15-1 design architecture consists of one input vector, 4 hidden layers with 15 neurons each and one output. Figure 6.10a, Fig.6.10b and Fig.6.10c illustrates the detailed architecture designs for compressive strength, split tensile strength and flexural strengths respectively.

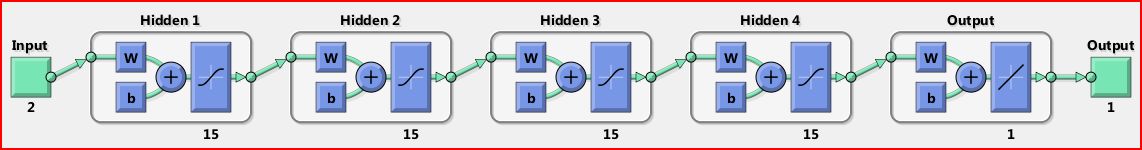


Fig. 6.9a GAANN 2-4-15-1 architecture for compressive strength

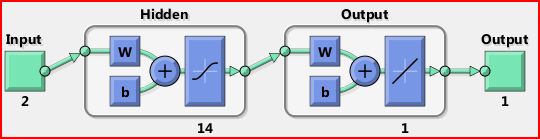


Fig. 6.9b GAANN 2-1-14-1 architecture for split tensile strength

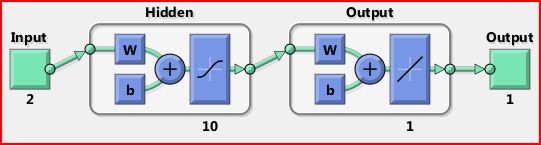


Fig. 6.9c GAANN 2-1-10-1 architecture for flexural strength

Hidden Neuron (j)

Hidden Neuron (k)

Bias = +1

Wkj

Bias = +1

Wkj

Bias = +1

Wkj

Output of Neuron

(Compressive Strength)

Output Neuron (n)

Bias = +1

Wlk

Input to Neuron(% fibre) and curing time

Input Neuron (i=2)

Bias = +1

Wji

Fig. 6.10a Proposed GAANN 2-4-15-1 design with two input, 4 hidden layers with 15 neurons each and one output for compressive strength

Output

One hidden layer with 10 nodes

Input layer

Days

% fibre fraction

Split tensile strength

Bias

1

2

13

14

Fig. 6.10b GAANN 2-1-14-1 architecture for Split tensile

Output

One hidden layer with 10 nodes

Input layer

Days

% fibre fraction

Flexural strength

bias

1

2

13

14

Fig. 6.10c GAANN 2-1-10-1 architecture for flexural strength

Similarly after examining both the numerical and graphical fit results, polynomial of degree 2 (p1x2 +p2x + p3) is selected as the best fit to extrapolate the compressive strength of SIFCON concrete. In the polynomial of degree 2, the R2 is less than one (≈ 0.99 in case of 8%, 10% fibre fraction and 0.97 in case of 12% fibre fraction) and confidence bounds do not cross zero on p1, p2 and p3.

Polynomial of degree 3 (p1x3 +p2x2 + p3x+p4) is selected as the best fit to extrapolate the split tensile and flexural strengths of SIFCON. The following sections present the comparative results of various ANN model designs and Polynomial curve fitting model with the LMANN model.

**6.8 Results and Discussion**

This section presents the possibilities of adopting Levenberg Marquardt back-propagation (LM) Artificial Neural Network (ANN), Steepest Descent Algorithm, Genetic Algorithm based ANN models and Curve fitting model using results of experimental investigation for prediction of SIFCON strength characteristics made with manufactured sand. Different percentage fibre fractions (8%, 10% and 12%) and various curing periods (7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91 and 98 days) have been provided as input vectors to these models. The designed networks have been trained with experimental data obtained from laboratory experimentation as presented in chapter 4.

For learning process, the input vectors and corresponding target vectors are used to train the LM, SD and Genetic models. The input data is corresponding to curing periods of SIFCON concrete, fibre configuration, number of neurons, learning rate, momentum and activation functions. At the initial stage of the experiment, data is scaled or normalized to within the range 0.1-0.9 using following equation

 (6.22)

where x new is the normalized value of an original data parameter, x is the original data point, xmin and xmax are the minimum and maximum values in the data set, respectively. This normalized form is chosen because it tends to provide a better outcome on the prediction of strength characteristic of SIFCON. The ratio of training to test data records employed in the experiment is 70:30. The test set should consist of a representative data set. The test set should be approximately 10-40% of the size of the training set of data.

**6.8.1 Performance Analysis of LMANN 2-2-15-1** **and Polynomial Curve Fitting model for compressive strength prediction**

The performance plot of the proposed LMANN 2-2-15-1 architecture is presented graphically in Fig. 6.11. This figure shows drop of the MSE as model learns i.e., as training progress. The blue line indicates the decrease in error on the training data whereas green line indicates the validation error and the neural network training stops when the validation error decreases. The red line indicates the error on the test data indicating how well the network generates the new data. This training of the network is stopped when the validation error decreased after 6 iterations. The results obtained are optimal because of the following reasons:

1. The final mean-squared error is small.
2. The test set error and the validation set error has similar characteristics.
3. No significant over fitting has occurred by iteration 14 (where the best validation performance occurs).

****

Fig. 6.11 Performance plot of the designed LMANN architecture

Tables 6.1, 6.2 and 6.3 illustrate the data sets that are used to model the LM based ANN and polynomial curve fitting method for predicting the compressive strengths of concrete for 8%, 10% and 12% fibre fractions respectively. The Mean Squared Error (MSE) obtained is 0.0101 for LM model and in case of curve fitting method the SSE obtained is 1.49.

Table 6.1 Predicted compressive strength for 8% fibre fraction using ANN and Curve Fitting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Days** | **% of fibre** | **Predicted compressive strength (MPa) using** | |
| **Curve Fitting**  **method** | **LM**  **Neural Networks**  **method** |
| 1 | 7 | 8 | 26.21 | 26.63 |
| 2 | 14 | 8 | 29.92 | 27.96 |
| 3 | 21 | 8 | 33.26 | 28.24 |
| 4 | 28 | 8 | 36.24 | 35.33 |
| 5 | 35 | 8 | 38.85 | 41.79 |
| 6 | 42 | 8 | 41.09 | 42.54 |
| 7 | 49 | 8 | 42.97 | 42.54 |
| 8 | 56 | 8 | 44.49 | 45.19 |
| 9 | 63 | 8 | 45.64 | 45.37 |
| 10 | 70 | 8 | 46.43 | 45.44 |
| 11 | 77 | 8 | 46.85 | 45.47 |
| 12 | 84 | 8 | 46.91 | 46.22 |
| 13 | 91 | 8 | 46.60 | 46.42 |
| 14 | 98 | 8 | 45.93 | 46.70 |

Table 6.2 Predicted compressive strength for 10% fibre using ANN and Curve Fitting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Days** | **Percent fibre** | **Predicted compressive strength (MPa) using** | |
| **Curve Fitting**  **method** | **LM Neural**  **Networks**  **method** |
| 1 | 7 | 10 | 35.48 | 35.29 |
| 2 | 14 | 10 | 37.89 | 35.92 |
| 3 | 21 | 10 | 40.07 | 40.98 |
| 4 | 28 | 10 | 42.05 | 42.68 |
| 5 | 35 | 10 | 43.84 | 42.91 |
| 6 | 42 | 10 | 45.46 | 42.93 |
| 7 | 49 | 10 | 46.94 | 46.08 |
| 8 | 56 | 10 | 48.29 | 47.15 |
| 9 | 63 | 10 | 49.52 | 51.51 |
| 10 | 70 | 10 | 50.67 | 51.69 |
| 11 | 77 | 10 | 51.75 | 52.88 |
| 12 | 84 | 10 | 52.77 | 53.25 |
| 13 | 91 | 10 | 53.76 | 53.32 |
| 14 | 98 | 10 | 54.74 | 53.77 |

Table 6.3 Predicted compressive strength for 12% fibre using ANN and Curve Fitting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Days** | **% of fibre** | **Predicted compressive strength (MPa) using** | |
| **curve Fitting method** | **LM Neural Networks**  **method** |
| 1 | 7 | 12 | 43.14 | 43.85 |
| 2 | 14 | 12 | 44.71 | 44.54 |
| 3 | 21 | 12 | 46.22 | 44.56 |
| 4 | 28 | 12 | 47.67 | 46.14 |
| 5 | 35 | 12 | 49.06 | 46.63 |
| 6 | 42 | 12 | 50.40 | 46.81 |
| 7 | 49 | 12 | 51.68 | 49.73 |
| 8 | 56 | 12 | 52.90 | 53.63 |
| 9 | 63 | 12 | 54.07 | 55.76 |
| 10 | 70 | 12 | 55.18 | 55.98 |
| 11 | 77 | 12 | 56.22 | 57.02 |
| 12 | 84 | 12 | 57.22 | 57.42 |
| 13 | 91 | 12 | 58.15 | 57.45 |
| 14 | 98 | 12 | 59.03 | 59.05 |

Table 6.4 Goodness of fit statistics for the SIFCON concrete

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **% fibre fraction** | **Polynomial degree** | **Equation** | **Polynomial Coefficients** | | | | **SSE** | **R2** | **RMSE** |
| **P1** | **P2** | **P3** | **P4** |
| 1 | 8% | 1 | p1x+p2 | 8.65 | 38.39 | - | - | 34.46 | 0.87 | 4.15 |
| 2 | 2 | p1x2 +p2x + p3 | -2.86 | 7.55 | 42.65 | - | 1.49 | 0.99 | 1.22 |
| 3 | 3 | p1x3 +p2x2 + p3x + p4 | -2.14 | -4.39 | 12.34 | 42.05 | 9.21e-28 | 1.00 | - |
| 4 | 10% | 1 | p1x+p2 | 7.47 | 44.61 | - | - | 5.32 | 0.97 | 1.63 |
| 5 | 2 | p1x2 +p2x + p3 | -1.59 | 7.84 | 45.80 | - | 1.63 | 0.99 | 1.36 |
| 6 | 3 | p1x3 +p2x2 + p3x + p4 | -2.40 | -2.19 | 4.96 | 45.84 | 9.09e-28 | 1.00 | - |
| 7 | 12% | 1 | p1x+p2 | 6.24 | 50.17 | - | - | 4.13 | 0.97 | 1.44 |
| 8 | 2 | p1x2 +p2x + p3 | -0.66 | 6.39 | 50.76 | - | 3.53 | 0.97 | 1.88 |
| 9 | 3 | p1x3 +p2x2 + p3x + p4 | -3.29 | 0.16 | 10.34 | 50.71 | 1.26e-27 | 1.00 | - |

Table 6.4 shows the Goodness of fit for the SIFCON concrete that has been obtained from the polynomial curve fitting method. This Table presents the various parameters that are obtained from the polynomial curve fitting method. The polynomial degree that has been used in this model is from 1 to 3. It is observed that even though R2 =1 is obtained for the third degree polynomial, the confidence bounds cross zero on p1 and p2 coefficients. This indicates that these coefficients (p1 and p2) may not differ from zero. However, in the case of polynomial degree of 2, it is observed that R2 is less than one (≈ 0.99 for 8%, 10% fibre fraction and 0.97 for 12% fibre fraction) and confidence bounds do not cross zero on p1, p2 and p3. This indicates that the fitted coefficients are fair and are accurately obtained. Therefore, after examining both the numerical and graphical fit results, polynomial of degree 2 is selected as the best fit to extrapolate the compressive strength of SIFCON concrete.

Figure 6.10a illustrates the variation of compressive strength of SIFCON concrete with 8% fibre fraction for different curing periods. Similarly Figs. 6.10b and 6.10c illustrate the variation for 10% and 12% fibre fraction. From the above figures, it is observed that the strength of SIFCON goes on increasing with the curing period. The variation in the rate of increase of the compressive strength is large at the early stage of curing period and later the rate of increase has declined. This is because of the fact that the hydration process gets slowed down. Also, it is observed that 12% fibre fraction has given the optimum strength when compared to other 8% and 10% fibre fractions.



Fig. 6.12a Variation of compressive strength (MPa) of SIFCON with curing period for 8% fibre fraction



Fig. 6.12b Variation of compressive strength (MPa) of SIFCON with curing period for 10% fibre fraction



Fig. 6.12c Variation of compressive strength (MPa) of SIFCON with curing period for 12% fibre fraction

Figure 6.13a illustrates the variation of compressive strength of SIFCON concrete for fibre fractions of 8%, 10% and 12% with different curing periods. It indicates that the strength goes on increasing with the increasing curing period. There is a large variation of increase in strength at the early stage of the curing period and later after 28 days of curing period the rate of increase of strength has started decreasing. This is because of the slowing down of the hydration process. Figure 6.13b illustrates the variation of compressive strength of SIFCON concrete for 12% fibre fraction for different curing periods. From Figure 6.13a and Fig. 6.13b, it is observed that 12% fibre fraction has given the optimum strength compared to the other fibre fractions (8% and 10% fibre fractions).



Fig. 6.13a Variation of compressive strength of SIFCON (fibre fraction of 8%, 10% and 12%) with different curing periods



Fig. 6.13b Variation of compressive strength of SIFCON for different curing periods with fibre fraction of 12%

A comparison was made between the two models i.e. Levenberg-Marquardt and polynomial curve fitting models in the above figures and it can be seen that LM model has shown better performance than that of polynomial curve fitting model. Even though the predicted compressive strengths due to curve fitting follow the same tendency as the predicted values due to ANN, the SSE and RMSE values of curve fitting are much higher than the MSEs obtained from the LM based ANN. It can therefore be seen that the Curve fitting is more applicable if the mathematical model of the process that produced the data is known.

**6.8.2 Performance Analysis of LMANN and SDANN for compressive strength prediction**

In this section, the performance of LM Algorithm and SD Algorithm based ANN models are compared in predicting the compressive strength of SIFCON concrete that contains cement, manufactured sand and various percentage fibre fractions, at different curing times. A total of 15 ANN architectures have been implemented and 2-4-14-1 architecture is proved to be the best suited architecture in the case of LM and 2-2-15-1 architecture in the case of SD algorithm. The optimized configuration provides a lower Root Mean Squares (RMS) Error. The LM requires less number of iterations when compared to SD, apart from high accuracy (95%), and fast convergence. The proposed ANN methods demonstrate that they are practical and beneficial to predict the strength of SIFCON. The results reveal that LM algorithm is more accurate than SD algorithm in predicting the compressive strength of SIFCON.

**Characteristics of SIFCON for different fibre configurations**:

The performance plot consisting of MSE, training errors, validation errors, and test errors of the proposed SDANN 2-4-14-1 architecture depicted in Fig. 6.8a is illustrated in Fig. 6.14a. This figure also shows the drop of MSE as the model learns i.e. as training progress. The blue line indicates the decrease in error on the training data whereas green line indicates the validation error and the neural network training stops when the validation error reduces. The red line shows the error on the test data indicating how well the network generates the new data. The obtained result is optimal because the final mean-square error is small, the test set error and the validation set error have similar characteristics and no significant over fitting has occurred by epoch 6 (where the best validation performance occurs).



Fig.6.14a Performance plot of the designed SDANN architecture

After successful completion of the training of the network, the network was subjected to predict the strength characteristics of SIFCON having different fibre percentages as input keeping the same curing times. The predicted compressive strengths are presented in Table 6.5.

The two algorithms were then compared and the comparison has been shown in Fig. 6.14b, where the percentage of fibre fraction is represented on horizontal axis and compressive strength on the vertical axis. Figure 6.14b shows variation of compressive strength of SIFCONat 7, 28, and 56 days of curing periods. The LM algorithm is represented by a dotted line, whereas, SD algorithm with a plain line. From Fig.6.14b it can be seen that the compressive strength predicted using LM algorithm is reasonably better than the SD algorithm. To highlight the LMANN performance more clearly, the predicted compressive strengths of SIFCON obtained using LM algorithm are presented graphically in Fig. 6.14c for the same curing periods.

Table 6.5 Predicted compressive strengths using LM and SD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Input** | | **Output (Compressive strength in MPa)** | |
| **Percentage**  **Fibre fraction (%)** | **Curing Period**  **(Days)** | **Levenberg-Marquardt** | **Steepest Descent** |
| 1 | S2 | 7 | 11.68 | 11.82 |
| 2 | S2 | 28 | 13.30 | 12.00 |
| 3 | S2 | 56 | 34.66 | 34.79 |
| 4 | S4 | 7 | 16.05 | 16.28 |
| 5 | S4 | 28 | 20.56 | 20.10 |
| 6 | S4 | 56 | 38.40 | 38.55 |
| 7 | S6 | 7 | 22.03 | 18.84 |
| 8 | S6 | 28 | 25.94 | 24.90 |
| 9 | S6 | 56 | 43.62 | 43.83 |
| 10 | S8 | 7 | 26.60 | 26.60 |
| 11 | S8 | 28 | 35.71 | 35.71 |
| 12 | S8 | 56 | 45.78 | 45.78 |
| 13 | S10 | 7 | 35.19 | 35.19 |
| 14 | S10 | 28 | 42.54 | 42.54 |
| 15 | S10 | 56 | 47.11 | 47.11 |
| 16 | S12 | 7 | 43.85 | 43.85 |
| 17 | S12 | 28 | 46.14 | 46.14 |
| 18 | S12 | 56 | 53.63 | 53.63 |
| 19 | S14 | 7 | 49.47 | 49.59 |
| 20 | S14 | 28 | 50.54 | 49.00 |
| 21 | S14 | 56 | 54.22 | 56.50 |
| 22 | S16 | 7 | 50.97 | 51.10 |
| 23 | S16 | 28 | 53.13 | 54.98 |
| 24 | S16 | 56 | 56.67 | 58.62 |
| 25 | S18 | 7 | 57.99 | 58.12 |
| 26 | S18 | 28 | 58.35 | 58.00 |
| 27 | S18 | 56 | 59.82 | 59.66 |
| 28 | S20 | 7 | 55.92 | 58.34 |
| 29 | S20 | 28 | 57.22 | 58.80 |
| 30 | S20 | 56 | 64.07 | 64.28 |
| 31 | S22 | 7 | 54.21 | 58.24 |
| 32 | S22 | 28 | 56.69 | 56.98 |
| 33 | S22 | 56 | 61.42 | 61.60 |

Table 6.5, shows the predicted compressive strength of SIFCON for various percentage fibre fractions ranging from 2% to 22%, at curing periods of 7, 28 and 56 days.

It is observed from the Table that, the compressive strength of SIFCON increased gradually as the fibre fraction increased from 2% to 18% and a decrease was observed in the compressive strength when fibre fraction increased beyond 18% till 22% at all curing periods. The decrease in the compressive strength was observed at of 7 and 28 days of curing, however, at 56 days of curing, the same drop was not observed. At 56 days, the strength has gradually increased when fibre volume fraction increased from 2% to 20% and decreased at 22% fibre volume.



Fig.6.14b Predicted compressive strength of SIFCON using LM and SD ANNs at 7, 28 and 56 days



Fig.6.14c Predicted compressive strength of SIFCON using LMANNs at 7, 28 and 56 days

The correlation coefficient and index of agreement between model output values and target values of the test data are used to evaluate the performance of the models. The other statistical parameters used to measure the predictive ability and stability of the models are the average, standard deviation (STD), and the minimum value of index of agreement. The larger the average computational time, the better will be the overall predictive ability of the model. The smaller the standard deviation, the higher will be the stability of the model. For example, the STD, average computation time, maximum error, minimum error and average error values observed for LM algorithm based ANN training set are 1.712, 5.12sec, 0.00123, 0.000312 and 0.001711 respectively and 8.0, 8.123sec, 0.5065, 0.037 and 0.2513 respectively for SD algorithms.

When the laboratory results are evaluated using LM algorithm and SD algorithm based ANNs; the predicted results of LMANN are better in comparison with SDANN. The accuracy achieved was 95% for LM algorithm whereas for SD algorithm it was 88%.

**Reasons for showing better performance by LM algorithm based ANN:**

Although LM algorithm requires more memory, it is accurate and the fastest supervised algorithm. In the SD algorithm, it is difficult to obtain a unique set of optimal parameters because of the existence of multiple local minima. The presence of these local minima slow down the search for global minimum because these algorithms frequently get trapped in local minima regions thereby incorrectly identify local minimum as the global minimum. The traditional conjugate gradient algorithm uses the gradient to compute a search direction followed by a line search algorithm to find the optimal step size along a line in the search direction.

The LM algorithm is a blend of the SD method employed in the back propagation algorithm and the quadratic rule employed in conjugate algorithms. The LM algorithm suggests that we should move further in the direction in which the gradient is smaller in order to get around the classic error valley.

**6.8.3 Performance Analysis of LM and GAANN for compressive strength prediction**

The performance plot consisting of MSE, training errors, validation errors, and test errors of the proposed GAANN 2-4-14-1 architecture is illustrated in Fig. 6.15, which also shows how the MSE drops as GAANN model learns. The blue curve indicates the decrease in error on the training data whereas green curve indicates the validation error and the neural network training stops when the validation error decreases. The red curve shows the error on the test data indicating how well the network generates the new data. The performance plot confirms that the proposed ANN design is optimal because the final MSE is small and the test set error and the validation set error have similar characteristics. No significant over fitting has occurred by iteration 2 (where the best validation performance 12.456 occurred).



Fig. 6.15 Performance plot of GAANN 2-4-14-1 architecture

The compressive strength of SIFCON cube specimens with different fibre volume fractions (2%, 4%, 6%, 8%, 10%, 12%, 14%, 16%, 18%, 20% and 22%) have been predicted using GAANN model from the values of compressive strengths of SIFCON corresponding to the fibre volume fractions 8%, 10% and 12% which were obtained from laboratory experimentation. Predicted values of compressive strengths of SIFCON for different fibre volume fractions were presented in Table 6.6.

From Table 6.6, it is observed that the predicted values have accuracy of 85%. The values of compressive strength obtained using GAANN has shown gradual increase in strength as the fibre volume fraction increases. Increase in strength has also been observed in cases where fibre volume fraction varied from 2% to 18%, whereas beyond 18 percent, a drop in the values of compressive strength was observed. Decrease in compressive strength is observed at 7 days and 28 days respectively. Whereas, at 56 days, specimens, a decrease in the compressive strength was observed at 22% fibre volume fraction (Fig. 6.16). Figure 6.16 shows the variation of predicted compressive strength for different percentage fibre volume fractions using the GAANN method at 7, 28 and 56 days of curing periods.

Table 6.6 Predicted compressive strengths of SIFCON for different fibre volume fractions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SIFCON specimen fibre**  **volume fraction**  **(%)** | **SIFCON cube specimens predicted Compressive strength (MPa) due to the proposed GAANN and their MSE values** | | | | | |
| **7 days specimen** | | **28days specimen** | | **56 days specimen** | |
| **Output** | **MSE** | **Output** | **MSE** | **Output** | **MSE** |
| S2 | 12.02 | 0.5648 | 14.57 | 2.0019 | 32.10 | 1.2325 |
| S4 | 15.65 | 0.9825 | 21.05 | 0.9954 | 36.97 | 0.9135 |
| S6 | 23.12 | 12.3602 | 26.34 | 1.7411 | 42.89 | 1.0046 |
| S8 | 26.30 | 0.0156 | 35.16 | 0.0552 | 45.39 | 0.4517 |
| S10 | 35.85 | 0.2250 | 43.22 | 4.1671 | 47.85 | 6.1414 |
| S12 | 42.65 | 0.2561 | 46.24 | 9.1121 | 52.89 | 0.5614 |
| S14 | 48.54 | 0.9423 | 51.66 | 3.4651 | 54.76 | 6.1411 |
| S16 | 51.77 | 0.4519 | 54.12 | 4.1654 | 55.81 | 26.4214 |
| S18 | 57.32 | 0.2354 | 59.03 | 1.7451 | 59.11 | 0.7844 |
| S20 | 54.96 | 6.2511 | 56.11 | 2.4121 | 63.51 | 1.2254 |
| S22 | 53.89 | 1.2377 | 55.32 | 1.4849 | 60.91 | 1.3348 |



Fig. 6.16 Variation of predicted compressive strengths of SIFCON using proposed GAANN with fibre volume fraction

**6.8.4 Performance Analysis of LM, SD, and GA based ANNs and Polynomial curve fitting algorithm for Split tensile and Flexural strengths prediction**

In the previous section (section 6.8.3) the results of compressive strengths obtained using different ANN based algorithms and polynomial curve fitting method are discussed. A similar approach has been made for the prediction of split tensile strength and flexural strength of SIFCON using LM, SD and GA based ANNs and also polynomial curve fitting method for different percentage fibre fractions and curing periods.

One of the major issues concerning the ANNs design is a proper adjustment of the weights of the network. There have been a number of studies comparing the performance of evolutionary and gradient based ANNs learning. But the results of the studies sometimes are conflicting to each other although the same standard data set development had been used. Three different training algorithms belonging to three classes (Steepest Descent, Levenberg-Marquardt and Genetic Algorithm) and Polynomial curve fitting were used to train ANN containing varying number of hidden layers and nodes in each layer combination. The ANNs were trained with those algorithms using the available experimental data as the training set. The divergence of the RMSE between the output and target values of test set was monitored and used as a criterion to stop training. The mean squared error function (MSE) and root mean squared error (RMSE= (MSE) 1/2) was used as the training error. The comparison between these algorithms is also made by considering the CPU time elapsed at the end of training. Precision and bias were evaluated through RMSE for test set and mean prediction error, respectively. The mean prediction error (MPE) or bias is calculated using the following equation

****

The parameters used for the genetic algorithm were mutation rate = 0.001, crossover rate = 0.7 with the selection method being uniform (suitably fit members are chosen to breed at random) and crossover method is single-point (parents genes are swapped over at one point along their length).

#### Table 6.7 and Table 6.8 shows the predicted split tensile strength of SIFCON obtained due to the optimum ANN model designs of LMANN 2-1-10-1 architecture, SDANN 2-2-14-1 architecture, GAANN 2-1-14-1 architecture and Polynomial of degree 3 curve fitting model using the experimental investigation data for various curing periods and different percentage of fibre fractions respectively.

#### Table 6.9 and Table 6.10 illustrates the Goodness of fit statistics for the split tensile of SIFCON for various curing periods and different percentage of fibre fractions respectively obtained from polynomial curve fitting method while extrapolating the strength results of the split tensile of SIFCON.

Table 6.11 and Table 6.12 illustrate the flexuralstrength ofSIFCON obtained for various curing periods and different percentage of fibre fractions respectively. These values are predicted from the optimum ANN model designs of LMANN 2-2-10-1 architecture, SDANN 2-2-14-1 architecture, GAANN 2-1-10-1 architecture and Polynomial of degree 3 curve fitting model using the experimental investigation data

From the analysis of the results presented in Tables 6.7, 6.8 and Tables 6.11 and 6.12 it is observed that the average RMSE for test set, training set, average mean prediction error, average number of epochs at the end of training and the average CPU time elapsed at the end of training observed in case of LMANN are 1.02495 (95% confidence bounds are: 0.8261 to 1.22377), 1.3318, 1.4820, 300 and 25 seconds. In case of SDANN, the average RMSE for test set, training set, average mean prediction error, average number of epochs at the end of training and the average CPU time elapsed at the end of training observed are 2.011075 (95% confidence bounds are: 1.79804 to 2.395575), 1.4818, 1.595525, 300 and 25 seconds. Similarly, the average RMSE for test set, training set, average mean prediction error, average number of epochs at the end of training and the average CPU time elapsed at the end of training observed in case of GAANN are 1.978075 (95% confidence bounds are: 1.7004 to 2.25575), 1.3318, 1.495525, 390 and 26 seconds.

From the results presented in the above Tables it can be seen that, LM based ANN prediction results have an accuracy of 96%, GA prediction has an accuracy of around 92%, SD algorithm based ANN obtained results have attained an accuracy of about 85% while polynomial curve fitting method have made up to 90%. Hence it is observed that the proposed LMANN model architecture has shown better performance or prediction accuracy than the remaining three predicting models.

Table 6.13 and Table 6.14 illustrates the Goodness of fit statistics obtained from polynomial curve fitting method while extrapolating the results of the flexural strength of SIFCON for various curing periods and different percentage fibre fractions respectively.

#### The values of the three static measures (Sum of Squares due to Error (SSE), RMSE and R-Square) used in assessing the performance of the curve fitting mode are presented i n Tables 6.9 and 6.10, 6.13 and 6.14. A value closer to zero indicates that the model has a smaller random error component, and that the fit will be more useful for prediction. SSE statistic measures the total deviation of the response values from the fit to the response values. The largest SSE obtained for split tensile and flexural strength of SIFCON indicates it is a poor fit. The lowest SSE value is associated with polynomial of degree 3. RMSE statistics is also known as the fit standard error and the standard error of the regression. It is an estimate of the standard deviation of the random component in the data. Similar to SSE, an MSE value closer to 0 indicates a fit that is more useful for prediction. The R-Square statistic measures how successful the fit is in explaining the variation of the data. R-square is defined as the ratio of the sum of squares of the regression (SSR) and the total sum of squares (SST). R-square can take on any value between 0 and 1, with a value closer to 1 indicating that a greater proportion of variance is accounted for by the model. For example, an R-square value of (polynomial degree 3) 0.98 means that the fit explains 98 % of the total variation in the data about the average. RMSE statistic is also known as the fit standard error and the standard error of the regression. It is an estimate of the standard deviation of the random component in the data. The R-square value of (polynomial degree 3) indicates that the fitted coefficients are fair and accurately known. Therefore, after examining both the numerical and graphical fit results, polynomial of degree 3 is selected as the best fit to extrapolate the compressive strength of SIFCON concrete. Even though, the R-square value of (polynomial degree 4) observed is 0.99, the confidence bounds on the coefficients that determine their accuracy cross zero for linear coefficients. This indicates that these coefficients (p1 and p2) may not differ from zero. Hence polynomial of degree 4 is not suitable for accurate prediction of strength characteristics of SIFCON.

Table 6.7 Split tensile strength prediction for different curing periods

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Percentage (%) fibre** | **Curing Period** | **Predicted split tensile Strength (MPa)** | | | |
| **LM** | **SD** | **GA** | **Curve Fitting** |
| 1 | 8 | 7 | 7.29 | 7.27 | 7.14 | 7.30 |
| 2 | 8 | 14 | 7.47 | 7.44 | 7.48 | 7.45 |
| 3 | 8 | 21 | 7.70 | 7.71 | 7.73 | 7.68 |
| 4 | 8 | 28 | 7.86 | 7.84 | 7.90 | 7.85 |
| 5 | 8 | 35 | 7.98 | 7.90 | 7.95 | 8.00 |
| 6 | 8 | 42 | 8.16 | 8.11 | 8.10 | 8.21 |
| 7 | 8 | 49 | 8.52 | 8.52 | 8.62 | 8.54 |
| 8 | 8 | 56 | 8.93 | 8.90 | 8.86 | 8.95 |
| 9 | 8 | 63 | 8.99 | 8.95 | 8.92 | 8.99 |
| 10 | 8 | 70 | 9.04 | 9.01 | 9.00 | 9.04 |
| 11 | 8 | 77 | 9.06 | 9.04 | 9.02 | 9.07 |
| 12 | 8 | 84 | 9.07 | 9.04 | 9.03 | 9.07 |
| 13 | 8 | 91 | 9.07 | 9.06 | 9.09 | 9.07 |
| 14 | 8 | 98 | 9.08 | 9.07 | 9.04 | 9.07 |
| 15 | 10 | 7 | 8.76 | 8.75 | 8.79 | 8.76 |
| 16 | 10 | 14 | 8.82 | 8.81 | 8.83 | 7.80 |
| 17 | 10 | 21 | 8.89 | 8.90 | 8.87 | 8.88 |
| 18 | 10 | 28 | 8.95 | 8.96 | 8.93 | 8.95 |
| 19 | 10 | 35 | 9.07 | 9.11 | 9.04 | 9.09 |
| 20 | 10 | 42 | 9.46 | 9.52 | 9.32 | 9.49 |
| 21 | 10 | 49 | 9.91 | 9.96 | 9.82 | 9.89 |
| 22 | 10 | 56 | 10.27 | 10.26 | 10.29 | 10.24 |
| 23 | 10 | 63 | 10.36 | 10.32 | 10.39 | 10.35 |
| 24 | 10 | 70 | 10.42 | 10.48 | 10.47 | 10.45 |
| 25 | 10 | 77 | 10.50 | 10.55 | 10.52 | 10.52 |
| 26 | 10 | 84 | 10.59 | 10.61 | 10.58 | 10.59 |
| 27 | 10 | 91 | 10.63 | 10.63 | 10.60 | 10.62 |
| 28 | 10 | 98 | 10.65 | 10.66 | 10.61 | 10.64 |
| 29 | 12 | 7 | 9.21 | 9.24 | 9.16 | 9.22 |
| 30 | 12 | 14 | 9.47 | 9.47 | 9.38 | 9.45 |
| 31 | 12 | 21 | 9.71 | 9.69 | 9.59 | 9.70 |
| 32 | 12 | 28 | 9.85 | 9.87 | 9.89 | 9.90 |
| 33 | 12 | 35 | 10.48 | 10.52 | 10.36 | 10.44 |
| 34 | 12 | 42 | 10.88 | 10.90 | 10.81 | 10.80 |
| 35 | 12 | 49 | 11.13 | 11.11 | 11.16 | 11.12 |
| 36 | 12 | 56 | 11.40 | 11.39 | 11.43 | 11.41 |
| 37 | 12 | 63 | 11.63 | 11.62 | 11.58 | 11.62 |
| 38 | 12 | 70 | 11.66 | 11.64 | 11.70 | 11.69 |
| 39 | 12 | 77 | 11.76 | 11.66 | 11.79 | 11.76 |
| 40 | 12 | 84 | 11.79 | 11.72 | 11.79 | 11.79 |
| 41 | 12 | 91 | 11.79 | 11.76 | 11.80 | 11.79 |
| 42 | 12 | 98 | 11.80 | 11.79 | 11.82 | 11.81 |

Table 6.8 Split tensile strength prediction for varying percentage fibre fractions

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Fibre percentage** | **Curing period** | **Predicted split tensile strength (MPa)** | | | |
| **LM** | **SD** | **GA** | **Curve Fitting** |
| 1 | 2 | 7 | 2.28 | 2.31 | 2.39 | 2.43 |
| 2 | 4 | 7 | 5.45 | 5.42 | 5.38 | 5.25 |
| 3 | 6 | 7 | 6.87 | 6.81 | 6.80 | 6.70 |
| 4 | 8 | 7 | 7.29 | 7.28 | 7.31 | 7.32 |
| 5 | 10 | 7 | 8.76 | 8.75 | 8.80 | 8.76 |
| 6 | 12 | 7 | 9.21 | 9.19 | 9.25 | 9.38 |
| 7 | 14 | 7 | 10.32 | 10.36 | 10.31 | 10.40 |
| 8 | 16 | 7 | 11.56 | 11.52 | 11.50 | 11.60 |
| 9 | 18 | 7 | 13.31 | 13.34 | 13.90 | 13.20 |
| 10 | 20 | 7 | 14.20 | 14.22 | 14.28 | 14.15 |
| 11 | 22 | 7 | 13.84 | 13.84 | 13.92 | 14.01 |
| 12 | 2 | 28 | 3.35 | 3.35 | 3.41 | 3.52 |
| 13 | 4 | 28 | 5.96 | 5.99 | 5.89 | 5.84 |
| 14 | 6 | 28 | 6.97 | 6.96 | 6.84 | 6.88 |
| 15 | 8 | 28 | 7.86 | 7.87 | 7.89 | 7.86 |
| 16 | 10 | 28 | 8.95 | 8.95 | 8.92 | 8.97 |
| 17 | 12 | 28 | 9.85 | 9.84 | 9.87 | 9.96 |
| 18 | 14 | 28 | 10.96 | 10.95 | 10.99 | 11.00 |
| 19 | 16 | 28 | 11.90 | 11.86 | 11.95 | 12.02 |
| 20 | 18 | 28 | 13.43 | 13.41 | 13.51 | 13.36 |
| 21 | 20 | 28 | 14.41 | 14.46 | 14.49 | 14.25 |
| 22 | 22 | 28 | 14.03 | 13.96 | 14.11 | 13.98 |
| 23 | 2 | 56 | 4.46 | 4.41 | 4.51 | 4.46 |
| 24 | 4 | 56 | 5.48 | 5.49 | 5.61 | 5.52 |
| 25 | 6 | 56 | 7.70 | 7.71 | 7.85 | 7.60 |
| 26 | 8 | 56 | 8.93 | 8.92 | 8.95 | 8.92 |
| 27 | 10 | 56 | 10.27 | 10.28 | 10.30 | 10.28 |
| 28 | 12 | 56 | 11.40 | 11.42 | 10.44 | 11.60 |
| 29 | 14 | 56 | 13.53 | 13.51 | 13.62 | 13.50 |
| 30 | 16 | 56 | 14.95 | 14.95 | 14.99 | 14.95 |
| 31 | 18 | 56 | 16.51 | 16.52 | 16.48 | 16.46 |
| 32 | 20 | 56 | 17.42 | 17.45 | 17.35 | 17.42 |
| 33 | 22 | 56 | 18.21 | 18.24 | 18.11 | 18.25 |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **% fibre fraction** | **Polynomial degree** | **Equation** | **Polynomial Coefficients** | | | | | **SSE** | **R2** | **RMSE** |
| **P1** | **P2** | **P3** | **P4** | **P5** |
| 1 | 8% | 1 | p1x+p2 | 0.02 | 7.29 | - | - | - | 0.58 | 0.90 | 0.21 |
| 2 | 2 | p1x2 +p2x + p3 | -2×10-4 | 0.05 | 6.84 | - | - | 0.21 | 0.96 | 0.13 |
| 3 | 3 | p1x3 +p2x2 + p3x + p4 | -5.02×10-6 | 5.61×10-4 | 1.17×10-2 | 7.19 | - | 0.10 | 0.98 | 0.10 |
| 4 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | 7.28×10-8 | -2.03×10-5 | 1.62×10-3 | -1.50×10-2 | 7.37 | 0.09 | 0.99 | 0.10 |
| 5 | 10% | 1 | p1x+p2 | 0.02 | 8.48 | - | - | - | 0.58 | 0.92 | 0.22 |
| 6 | 2 | p1x2 +p2x + p3 | -1.2×10-4 | 0.04 | 8.23 | - | - | 0.47 | 0.93 | 0.21 |
| 7 | 3 | p1x3 +p2x2 + p3x + p4 | -8.55×10-6 | 1.21×10-3 | -0.02 | 8.828 | - | 0.17 | 0.97 | 0.13 |
| 8 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | 1.64×10-7 | -4.30×10-5 | 3.60×10-3 | -0.08 | 9.24 | 0.11 | 0.99 | 0.11 |
| 9 | 12% | 1 | p1x+p2 | 0.03 | 9.25 | - | - | - | 1.22 | 0.89 | 0.31 |
| 10 | 2 | p1x2 +p2x + p3 | -3.75×10-4 | 0.07 | 8.52 | - | - | 0.23 | 0.98 | 0.14 |
| 11 | 3 | p1x3 +p2x2 + p3x + p4 | -4.86×10-6 | 3.91×10-4 | 0.04 | 8.86 | - | 0.14 | 0.98 | 0.11 |
| 12 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | 1.87×10-7 | -4.41×10-5 | 3.10×10-3 | -0.01 | 9.33 | 0.05 | 0.99 | 0.07 |

Table 6.9 Goodness of fit statistics for the split tensile of SIFCON for various curing periods

Table 6.10 Goodness of fit statistics for the split tensile of SIFCON for different fibre fractions

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **% fibre fraction** | **Polynomial degree** | **Equation** | **Polynomial Coefficients** | | | | | **SSE** | **R2** | **RMSE** |
| P1 | P2 | P3 | P4 | P5 |
| 1 | 8% | 1 | p1x+p2 | 0.55 | 2.70 | - | - | - | 5.34 | 0.96 | 0.77 |
| 2 | 2 | p1x2 +p2x + p3 | -0.1 | 0.81 | 1.57 | - | - | 3.71 | 0.97 | 0.68 |
| 3 | 3 | p1x3 +p2x2 + p3x + p4 | 7.90×10-4 | -3.93×10-2 | 1.10 | 0.88 | - | 3.46 | 0.98 | 0.70 |
| 4 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -5.37×10-4 | 0.02 | -0.45 | 3.52 | -3.14 | 0.41 | 0.99 | 0.26 |
| 5 | 10% | 1 | p1x+p2 | 0.53 | 3.42 | - | - | - | 3.17 | 0.97 | 0.59 |
| 6 | 2 | p1x2 +p2x + p3 | -9.82×10-3 | 0.76 | 2.40 | - | - | 1.85 | 0.98 | 0.48 |
| 7 | 3 | p1x3 +p2x2 + p3x + p4 | 1.58×10-4 | -0.02 | 0.82 | 2.26 | - | 1.84 | 0.98 | 0.51 |
| 8 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -3.89×10-4 | 0.02 | -0.31 | 2.57 | -0.64 | 0.24 | 0.99 | 0.20 |
| 9 | 12% | 1 | p1x+p2 | 0.72 | 3.08 | - | - | - | 1.42 | 0.99 | 0.39 |
| 10 | 2 | p1x2 +p2x + p3 | -5.3×10-3 | 0.84 | 2.53 | - | - | 1.03 | 0.99 | 0.35 |
| 11 | 3 | p1x3 +p2x2 + p3x + p4 | -8.99×10-4 | 0.02 | 0.5 | 3.31 | - | 0.71 | 0.99 | 0.31 |
| 12 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -1.26×10-4 | 5.14×10-3 | -0.06 | 1.09 | 2.37 | 0.54 | 0.99 | 0.30 |

Table 6.11 Flexural strength prediction (for different curing periods)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Percentage fibre** | **Curing Period** | **Predicted Flexural Strength (MPa)** | | | |
| **LM** | **SD** | **GA** | **Curve fitting** |
| 1 | 8 | 7 | 21.71 | 21.69 | 20.96 | 21.70 |
| 2 | 8 | 14 | 22.31 | 22.05 | 21.62 | 22.58 |
| 3 | 8 | 21 | 23.51 | 23.86 | 22.96 | 23.43 |
| 4 | 8 | 28 | 23.95 | 23.98 | 23.35 | 23.89 |
| 5 | 8 | 35 | 24.11 | 24.19 | 23.99 | 24.15 |
| 6 | 8 | 42 | 24.50 | 24.63 | 24.12 | 24.56 |
| 7 | 8 | 49 | 24.92 | 25.03 | 24.91 | 24.94 |
| 8 | 8 | 56 | 25.22 | 25.25 | 25.06 | 25.28 |
| 9 | 8 | 63 | 25.99 | 26.08 | 25.86 | 25.88 |
| 10 | 8 | 70 | 26.34 | 26.26 | 26.01 | 26.28 |
| 11 | 8 | 77 | 26.56 | 26.49 | 26.23 | 26.56 |
| 12 | 8 | 84 | 26.74 | 26.81 | 26.59 | 26.76 |
| 13 | 8 | 91 | 26.91 | 26.98 | 26.82 | 26.93 |
| 14 | 8 | 98 | 27.11 | 27.16 | 27.00 | 27.09 |
| 15 | 10 | 7 | 26.91 | 26.93 | 26.22 | 26.95 |
| 16 | 10 | 14 | 27.09 | 27.11 | 26.98 | 27.11 |
| 17 | 10 | 21 | 27.16 | 27.21 | 27.11 | 27.19 |
| 18 | 10 | 28 | 27.46 | 27.49 | 27.32 | 27.44 |
| 19 | 10 | 35 | 27.77 | 27.62 | 27.42 | 27.78 |
| 20 | 10 | 42 | 27.80 | 27.86 | 27.71 | 28.04 |
| 21 | 10 | 49 | 28.87 | 28.92 | 28.09 | 28.72 |
| 22 | 10 | 56 | 29.16 | 29.14 | 28.89 | 29.14 |
| 23 | 10 | 63 | 29.53 | 29.56 | 29.28 | 29.49 |
| 24 | 10 | 70 | 29.67 | 29.62 | 29.55 | 26.67 |
| 25 | 10 | 77 | 29.77 | 29.80 | 29.82 | 29.80 |
| 26 | 10 | 84 | 29.89 | 29.90 | 29.95 | 29.93 |
| 27 | 10 | 91 | 30.01 | 30.01 | 30.10 | 30.02 |
| 28 | 10 | 98 | 30.08 | 30.06 | 30.88 | 30.05 |
| 29 | 12 | 7 | 31.75 | 31.79 | 31.23 | 31.85 |
| 30 | 12 | 14 | 33.56 | 33.59 | 32.88 | 33.52 |
| 31 | 12 | 21 | 34.84 | 34.80 | 33.95 | 34.62 |
| 32 | 12 | 28 | 35.67 | 35.66 | 34.81 | 35.59 |
| 33 | 12 | 35 | 35.99 | 36.01 | 35.75 | 36.00 |
| 34 | 12 | 42 | 36.36 | 36.42 | 36.10 | 36.40 |
| 35 | 12 | 49 | 36.95 | 36.99 | 36.51 | 36.89 |
| 36 | 12 | 56 | 37.44 | 37.41 | 36.88 | 37.46 |
| 37 | 12 | 63 | 37.94 | 37.99 | 37.68 | 37.92 |
| 38 | 12 | 70 | 38.22 | 38.21 | 38.55 | 38.20 |
| 39 | 12 | 77 | 38.52 | 38.62 | 38.81 | 38.52 |
| 40 | 12 | 84 | 38.95 | 38.98 | 38.91 | 38.90 |
| 41 | 12 | 91 | 39.06 | 39.10 | 39.01 | 39.05 |
| 42 | 12 | 98 | 39.09 | 39.11 | 39.05 | 39.10 |

Table 6.12 Flexural strength prediction (for varying percentage fibre fractions)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Percentage fibre** | **Curing period** | **Flexural strength (MPa)** | | | |
| **LM** | **SD** | **GA** | **Curve Fitting** |
| 1 | 2 | 7 | 6.48 | 6.56 | 6.38 | 6.50 |
| 2 | 4 | 7 | 11.93 | 11.48 | 10.79 | 11.90 |
| 3 | 6 | 7 | 16.98 | 16.76 | 17.05 | 16.95 |
| 4 | 8 | 7 | 21.71 | 21.62 | 21.54 | 21.74 |
| 5 | 10 | 7 | 26.91 | 26.83 | 26.62 | 26.93 |
| 6 | 12 | 7 | 31.75 | 31.79 | 31.89 | 31.75 |
| 7 | 14 | 7 | 35.98 | 34.43 | 33.66 | 35.90 |
| 8 | 16 | 7 | 39.06 | 38.89 | 38.42 | 39.00 |
| 9 | 18 | 7 | 40.68 | 39.96 | 41.21 | 40.72 |
| 10 | 20 | 7 | 42.41 | 42.21 | 43.07 | 42.45 |
| 11 | 22 | 7 | 45.23 | 45.04 | 45.12 | 45.20 |
| 12 | 2 | 28 | 8.92 | 9.35 | 8.24 | 8.92 |
| 13 | 4 | 28 | 13.45 | 13.38 | 13.86 | 13.47 |
| 14 | 6 | 28 | 19.96 | 20.14 | 21.23 | 19.90 |
| 15 | 8 | 28 | 23.95 | 23.98 | 23.77 | 23.95 |
| 16 | 10 | 28 | 27.46 | 27.41 | 27.32 | 27.56 |
| 17 | 12 | 28 | 35.67 | 35.71 | 35.42 | 35.64 |
| 18 | 14 | 28 | 40.48 | 40.24 | 39.12 | 40.46 |
| 19 | 16 | 28 | 43.14 | 43.89 | 43.51 | 43.20 |
| 20 | 18 | 28 | 48.56 | 48.96 | 47.54 | 48.56 |
| 21 | 20 | 28 | 51.91 | 51.63 | 51.68 | 51.86 |
| 22 | 22 | 28 | 50.23 | 50.29 | 49.77 | 50.25 |
| 23 | 2 | 56 | 10.69 | 10.79 | 10.21 | 10.75 |
| 24 | 4 | 56 | 15.91 | 16.12 | 16.07 | 15.89 |
| 25 | 6 | 56 | 21.97 | 21.69 | 22.15 | 20.98 |
| 26 | 8 | 56 | 25.22 | 25.18 | 25.42 | 25.22 |
| 27 | 10 | 56 | 29.16 | 29.20 | 29.31 | 30.20 |
| 28 | 12 | 56 | 37.14 | 37.09 | 37.05 | 37.16 |
| 29 | 14 | 56 | 43.91 | 43.97 | 44.22 | 43.90 |
| 30 | 16 | 56 | 47.68 | 47.89 | 46.98 | 47.70 |
| 31 | 18 | 56 | 52.14 | 52.31 | 52.54 | 52.14 |
| 32 | 20 | 56 | 56.31 | 56.21 | 56.84 | 55.62 |
| 33 | 22 | 56 | 52.91 | 52.86 | 53.92 | 53.20 |

Table 6.13 Goodness of fit statistics of flexural strength of SIFCON for various curing periods

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **% fibre fraction** | **Polynomial degree** | **Equation** | **Polynomial Coefficients** | | | | | **SSE** | **R2** | **RMSE** |
| **P1** | **P2** | **P3** | **P4** | **P5** |
| 1 | 8% | 1 | p1x+p2 | 0.06 | 21.96 | - | - | - | 0.88 | 0.95 | 0.28 |
| 2 | 2 | p1x2 +p2x + p3 | -3.94×10-4 | 0.009918 | 21.18 | - | - | 0.50 | 0.98 | 0.21 |
| 3 | 3 | p1x3 +p2x2 + p3x + p4 | 1.74×10-6 | -6.69×10-4 | 0.1111 | 21.06 | - | 0.48 | 0.98 | 0.22 |
| 4 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -2.64×10-7 | 5.79×10-5 | -4.5×10-3 | 0.20 | 20.4 | 0.32 | 0.99 | 0.19 |
| 5 | 10% | 1 | p1x+p2 | 0.04 | 26.53 | - | - | - | 1.10 | 0.94 | 0.30 |
| 6 | 2 | p1x2 +p2x + p3 | -1.75×10-4 | 0.06 | 23.18 | - | - | 0.88 | 0.95 | 0.28 |
| 7 | 3 | p1x3 +p2x2 + p3x + p4 | -1.12×10-5 | 1.5×10-3 | -0.017 | 26.97 | - | 0.36 | 0.98 | 0.19 |
| 8 | 4 | p1x3 +p2x2 + p3x + p4+ p5 | 1.59×10-7 | -4.47×10-5 | 3.90×10-3 | -0.07 | 27.37 | 0.31 | 0.99 | 0.18 |
| 9 | 12% | 1 | p1x+p2 | 0.07 | 32.96 | - | - | - | 5.84 | 0.90 | 0.69 |
| 10 | 2 | p1x2 +p2x + p3 | -8.10×10-4 | 0.15 | 31.37 | - | - | 1.25 | 0.98 | 0.33 |
| 11 | 3 | p1x3 +p2x2 + p3x + p4 | 1.15×10-5 | -2.62×10-3 | 0.23 | 30.57 | - | 0.71 | 0.98 | 0.26 |
| 12 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -4.9×10-7 | 1.15×10-4 | -9.8×10-3 | 0.41 | 29.32 | 0.14 | 0.99 | 0.12 |

Table 6.14 Goodness of fit statistics of flexural strength of SIFCON for different fibre fractions

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **% fibre fraction** | **Polynomial degree** | **Equation** | **Polynomial Coefficients** | | | | | **SSE** | **R2** | **RMSE** |
| **P1** | **P2** | **P3** | **P4** | **P5** |
| 1 | 8% | 1 | p1x+p2 | 1.95 | 5.52 | - | - | - | 52.54 | 0.96 | 2.41 |
| 2 | 2 | p1x2 +p2x + p3 | -0.06 | 3.38 | -0.65 | - | - | 3.99 | 0.98 | 0.70 |
| 3 | 3 | p1x3 +p2x2 + p3x + p4 | -1.64×10-3 | -1.95×10-4 | 2.79 | 0.78 | - | 2.92 | 0.99 | 0.64 |
| 4 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | 3.56×10-4 | -1.87×10-2 | 0.27 | 1.18 | 3.44 | 1.58 | 0.99 | 0.51 |
| 5 | 10% | 1 | p1x+p2 | 2.26 | 5.92 | - | - | - | 56.24 | 0.97 | 2.5 |
| 6 | 2 | p1x2 +p2x + p3 | -.04 | 3.37 | 1.08 | - | - | 26.47 | 0.98 | 1.81 |
| 7 | 3 | p1x3 +p2x2 + p3x + p4 | -5.93×10-3 | 0.16 | 1.23 | 6.26 | - | 12.56 | 0.99 | 1.33 |
| 8 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -5.40×10-4 | 0.02 | -0.24 | 3.67 | 2.21 | 9.47 | 0.99 | 1.25 |
| 9 | 12% | 1 | p1x+p2 | 2.37 | 7.21 | - | - | - | 76.99 | 0.97 | 2.92 |
| 10 | 2 | p1x2 +p2x + p3 | -0.04 | 3.44 | 2.56 | - | - | 49.63 | 0.98 | 2.49 |
| 11 | 3 | p1x3 +p2x2 + p3x + p4 | -8.8×10-3 | 0.27 | 0.27 | 10.26 | - | 18.98 | 0.99 | 1.64 |
| 12 | 4 | p1x3 +p2x2 + p3x + p4x + p5 | -9.86×10-4 | 0.04 | -0.48 | 4.72 | 2.87 | 8.72 | 0.99 | 1.20 |

**6.9 Conclusions**

In this chapter, the application of Levenberg-Marquardt, SD, Genetic based neural network models and Polynomial Curve fitting for predicting the strength characteristics of SIFCON has been demonstrated. It has been concluded that the addition of steel fibre has significantly increased the strengths of SIFCON. In all the proposed ANN models, the values of compressive strength of SIFCON obtained has gradually increased with the curing period i.e., from 7 to 91 days. On comparison of addition of 8%, 10% and 12% steel fibres in SIFCON, 12% fibre addition has shown maximum value of compressive strength at all the curing periods. Whereas at 56 days of curing, gradual increase in the compressive strength was obtained as fibre volume was increased from 2% to 20%. But when the fibre percentage is increased beyond 20%, a slight drop is observed in the compressive strength. LMANN model performance has also been compared with all other models. At 7 days of curing, the maximum MSE of the estimated compressive strength was observed as 9.3428 (at 6% fibre fraction) and the minimum MSE was observed as zero (at 12% fibre fraction). Whereas, the maximum MSE observed at 28 days and 56 days of curing were 7.3095 (at 12%) and 23.3801 (at 16%) respectively; and the minimum MSE observed for 28 days and 56 days of curing were 0.0011 (at 8%) and 0.0002 (at 2%) respectively.

The LM, SD, the polynomial curve fitting model and Genetic ANN model are also able to predict the split tensile and flexural strengths of SIFCON concrete satisfactorily with an accuracy of about 96%, 85%, 90% and 92% respectively. Thus, it is concluded that the LM algorithm gives better performance and accuracy than that of SD and GANN algorithm and Curve Fitting method. Therefore, it can be concluded that the developed neural network model architectures are best suited for predicting the strength characteristics of SIFCON made with manufactured sand use of low strength steel binding wire as fibre.