5th Conference on Sustainability in Civil Engineering (CSCE'23)
Department of Civil Engineering
Capital University of Science and Technology, Islamabad Pakistan

STRENGTH PREDICTION OF VARIOUS BEAMS THROUGH THE ARTIFICIAL NEURAL NETWORK

[a] Muhammad Mahtab Ahmad*, [b] Ayub Elahi

a: Department of Civil Engineering, UET, Taxila, Pakistan, mahtabahmad010@gmail.com
b: Department of Civil Engineering, UET, Taxila, Pakistan, ayub.elahi@uettaxila.edu.pk
* Corresponding author: Email ID: mahtabahmad010@gmail.com

Abstract- This study aims to compare of first crack load and flexural strength of reinforced concrete beams without stirrups obtained from the conventional model developed using the current design code (ACI building code) with the non-conventional problem solver, i.e., an Artificial Neural Network (ANN). For this purpose, 110 sample data of reinforced concrete beams without stirrups reinforcement obtained from published research data are used to train Multilayer Backpropagation Neural Network through MATLAB. This work enables the development of a knowledge-based structural analysis model capable of predicting RC structural responses. The results obtained from the ANN model are closer to the experimental results of the conventional model. The coefficient of determination obtained from the comparison of these results is 0.945.

Keywords- Reinforced Concrete, Soft Computing, Artificial Neural Network, Ultimate Limit State, Finite Element, Multilayer Back Propagation, Nonlinear Finite Element Analysis, The Central Nervous System,

1 Introduction

Researchers and engineers have proposed essential theories [1–3] and techniques [4] to accurately forecast the behaviour of reinforced concrete (RC) structure elements at the ultimate limit state (ULS), ensuring safety and cost-effectiveness. Soft Computing (SC) methods [5–7], such as Artificial Neural Networks (ANNs) and Genetic Algorithms [8], have emerged as powerful computational tools that deviate from traditional analysis procedures and offer improved accuracy. ANNs provide precise and economical solutions with minimal analysis time, making them more efficient than conventional numerical procedures like the finite element (FE) method. To analyse RC structures effectively, accurate prediction of the nonlinear behaviour of individual components is crucial. ANNs rely on heuristic approaches rather than strict mechanics, and their calibration process must consider critical factors, including essential design parameters [9]. Experimental studies using scaled models [2, 10, 11] provide the database for the ANN model, capturing the behaviour and load-carrying capacity of RC members. However, the input values from these studies may not represent the design parameters used in full-scale RC members, limiting the precision of ANN predictions. Nonlinear Finite Element Analysis (NLFEA) and physical models using RC design codes [2, 10] [12] can offer valuable insights but may require re-calibration and have limitations in predicting full-scale RC members [13]. The main objective of this research is to train an ANN model to accurately predict structural properties (e.g., flexural strength, shear strength, and first crack load value) of RC beams based on key design parameters approaching the ULS. The study utilizes Multilayer Back Propagation (MBP) Neural Networks and MATLAB [12, 13] to develop an open-source analysis tool allowing users to modify parameters and solve a broader range of engineering problems. The proposed framework comprises several components: (1) analysis of relevant test data to create databases for developing the ANN model, (2) focus on the architectural formation of the ANN model, (3) training the ANN model, and (4) developing a function to expand the ANN model's application for predicting RC structure response, even in cases where available experimental databases lack design parameters or inadequately represent them [16].

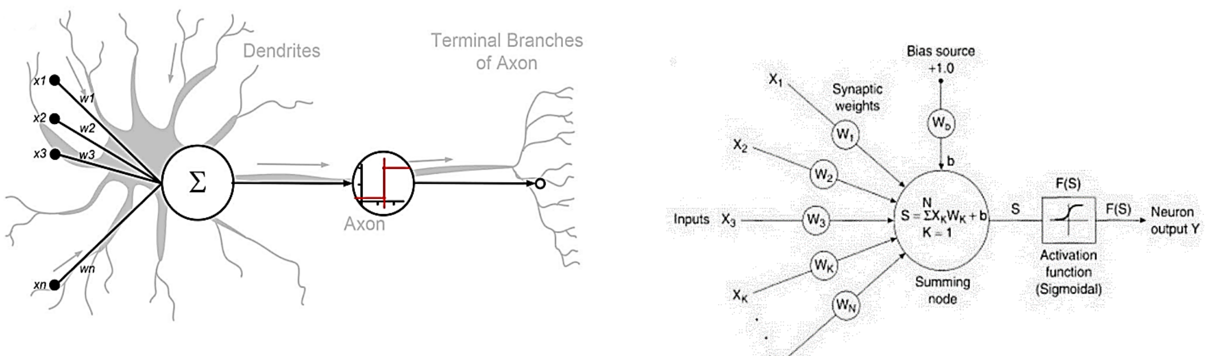
1.1 Artificial Neural Network Morphology

Artificial Neural Networks (ANNs) are computational models inspired by the biological neural networks found in human and animal brains. They are designed to process information, learn from it, identify patterns, and make predictions, like the biological nervous system. ANNs have a wide range of applications in natural language processing, image recognition, and data prediction. The structure of an ANN consists of interconnected neurons organized into layers, including an input layer, one or more hidden layers, and an output layer. Each neuron is connected to others through weighted links. These weights determine the significance of the input values. The values from the neurons in each layer are summed with bias values, and this process is repeated for subsequent layers, forming a network that processes and transmits information. Activation functions define the relationship between neurons in consecutive layers. The output of a neuron in one layer becomes the input for the neurons in the next layer. Initially, the weights are assigned randomly, but during the training process, iterations are performed to adjust the weight values to achieve accurate output predictions. Equation (1 & 2) provides an analytical expression for the summation of weights [20].

$x_j = \sum(y_i W_{ji}) + b_j$ Equation (1)

$y_j = f(x_j)$ Equation (2)

In the above Equations, x_j represents the output from a specific neuron, y_k represents the results received from applying the activation function $f(x_j)$, W_{ji} represent weights coefficients used between interconnected neuron, b_j is the bias value for the neuron, and "j" and "i" represent the number of layers and neurons in each network, respectively.



a b

Figure 1 a) Biological and Artificial Neural Network Morphology, b) Structure of ANN model[20]

2 Artificial Neural Network Model Formation

2.1 Multilayered Backpropagation ANN Model Structure

The Multilayered Backpropagation Neural Network (MBNN) model is widely used to predict the behaviour of reinforced concrete structures [19]. The visual representation of MBNN is shown in Figure 3 (b). It operates in two phases: free forward calculation and Error Signal Backpropagation. In the first phase, input parameters from the sample database are fed into the input layer neurons. Weights are multiplied with inputs, and biases are summed to produce neuron outputs. An activation function introduces nonlinearity. This process continues through hidden layers, and their outputs serve as inputs for the output layer, generating predictions. Predicted results are compared with target values in the database, calculating the error [21, 22]

$E = \frac{1}{2}\sum(X_j - Y_j)^2$ Equation (3)

The error signal obtained from the previous phase is used to adjust the weights and biases of the MBNN. This involves changing the network architecture, such as the number of neurons or layers, and increasing the cycles. The feedforward process is repeated to minimize errors and optimize the network for accurate results [20]. The "Gradient Descent" method is employed, where weight adjustments depend on the error function "E" and learning rate "η" [16]. Correction is calculated using equation (4).

$$\Delta W_{ji} = -\eta \frac{\delta E}{\delta W_{ji}} \quad \text{Equation (4)}$$

As from equation 4, if a higher learning rate value is applied, then abrupt changes in the value of weights would come out during each iteration if higher values of weights are used initially; this would lead to a prolonged process to achieve an optimised neural network model. On the other hand, with a small value of learning rate and small values of weights initially applied, training cycles are increased with prolonged training time but can proceed to a more optimised Neural Network model.

2.2 Transfer Function

Activation functions in an ANN model play a vital role in processing and transmitting information between layers. The choice of activation function depends on the problem and input parameter normalization. Different functions are applied between layers. For instance, the "log-sigmoid" function is used between the input and hidden layers when data is normalized between 0 and 1. When normalization yields result between -1 and 1, the "linear" function is used for the last two layers. In decision-making cases, the "hyperbolic tangent" function can be used between all layers, with "Gaussian" for the output layer and "hyperbolic tangent" for the hidden layer. Activation function selection has no strict rule, and each has advantages and limitations. Sigmoid provides non-linearity and an output range of 0 to 1 but faces vanishing gradients and output saturation. Tanh introduces non-linearity, has zero-centred output, stronger gradients, and vanishing gradients and no finite bounds. ReLU is popular in deep learning due to sparsity, efficiency, and avoiding vanishing gradients, but can lead to dead neurons and lacks an upper bound. Softmax is used in multi-class classification for class probabilities but can be sensitive to large inputs and assumes class independence. Consider these factors to select appropriate activation functions based on task and network architecture, ensuring optimal performance and stability in the ANN model.

2.3 MBNN Model Creation

For predicting flexural strength, shear strength, and first crack load in beams, the Multilayer Back Propagation neural network (MBPNN) is selected. Previous studies indicate that the network's performance depends on factors such as sample database size, number of hidden layers and neurons, and initial weights and biases. Activation functions, error functions, and learning algorithms are also critical in determining the learning rate and performance. Literature suggests optimizing the neural network for improved performance and fast learning rate [20, 23],

1. Initial values for weights and biases should be assigned between -0.5 and 0.5.
2. In the hidden layer number of neurons should be double the amount neurons in the input layer
3. The activation function for the first two layers (input layer and hidden layer) should use used sigmoid activation function, while the output layer hyperbolic tangent activation function should be used.

Determining the ideal number of hidden layers and neurons in an artificial neural network is challenging and crucial for its predictive capabilities. Increasing layers or neurons can improve pattern learning but may be overfitted without proper regularization. Reducing complexity can lead to underfitting. Cross-validation and grid search techniques are used to find optimal configurations. Cross-validation divides the dataset, and trains and evaluates the network multiple times, comparing performance across configurations. Grid search explores hyperparameter combinations to select the best configuration based on validation set performance. Both approaches aim for a balanced architecture maximizing prediction accuracy. Optimal architecture depends on the dataset and problem, requiring careful evaluation and experimentation.

2.4 Sample Database

110 Sample databases were included in this network training testing and validation process. Data is based on a simply supported beam without shear reinforcement, whose sampling was done in UET Taxila concrete laboratory. The ranges of the database are listed in the table below.

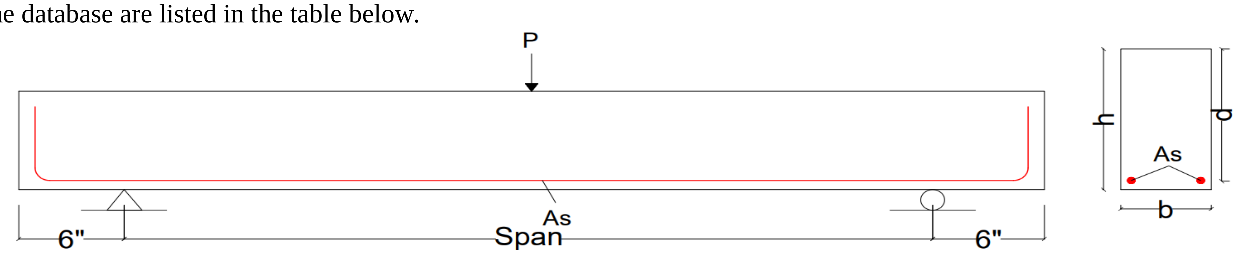


Figure 2 Beam cross Section details.

Table 1 Range of sample database for network training

	b	d	Av/d	ρ_1	a_1	fy	fc	L	M_f	P_{cr}
Units	**inch**	**inch**		**%**		**psi**	**psi**	**inch**	**lb.in**	**lb.**
Max	7	12.5	6	2.057	2.91	77222	8339	126	1047084	54574
Min	6	10.5	1	0.349	0.37	70727	7697	21	95400	2160
SD	0.502	1.004	1.588	0.582	0.768	2599.74	157.47	33.35	237180.73	10268.58
Avg.	6.5	11.5	3.5	1.099	1.389	74286	8024.29	85.5	483318.08	12606.42
COV	0.077	0.087	0.454	0.529	0.553	0.035	0.0196	0.39	0.49	0.815

2.5 Data Processing

ANN performance relies on the quality of the database. To optimize the network, it's crucial to normalize the database for efficient training. After training, data can be denormalized for comparison with target values. Normalization converts values to unitless form, ensuring consistency. Database normalization between upper and lower values prevents slow learning. MATLAB's Neural Network Toolbox provides functions for normalization. Performing normalization before data loading allows for greater control. In the mentioned model, data is normalized between 0.1 and 1 using equation (5).

$$x' = \frac{(x-x_{min})}{(x_{max}-x_{min})}(u-l)+l \qquad \text{Equation (5)}$$

Where x' is the new normalized value , x is the original value, u is the upper limit for normalization,and l is the lower limit for normalization.

2.6 Division of Database

Data is split into three subsets: training, validation, and testing. The training subset is used for weight and bias adjustments through gradient updates. The validation subset calculates the error function and supports backpropagation during training. The testing subset evaluates different network models and compares errors with the validation process. If discrepancies arise, the dataset division is adjusted to minimize differences. Our model assigns 70% for training, 15% for validation, and 15% for testing. MATLAB's "dividerand" command randomizes the database division for optimized efficiency.

2.7 The Functionality of ANN Model

ANN model is trained as a multi-layer model that is coded in MATLAB using the Levenberg-Marquardt algorithm with the free forward back-propagation method [4, 19, 20]. The key aspects of training are summarized as

1) The training process involves dividing the database into three subsets using a random method. 70% of the data is used for training, 15% for validation, and the remaining 15% for testing.
2) 1000 epochs/cycle are selected to train the ANN model, and the training is stopped if either of the following conditions is met: (a) a maximum of 100 validation failures occur, or (b) the minimum performance learning slope becomes 10^{-8}.
3) The error value of the correlation factor (R), the mean absolute error (MAE), and the mean squared error (MSE) are used to select the optimized ANN model [21–23]. These measures are expressed analytically by equations (6)-(8) respectively, as referenced from the literature.

$$R = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sum_{i=1}^{n}(X_i-\bar{X})^2\sum_{i=1}^{n}(Y_i-\bar{Y})^2} \qquad \text{Equation (6)}$$

$$MSE = \frac{\sum_{i=1}^{n}({X_i}^2-{Y_i}^2)}{n} \qquad \text{Equation (7)}$$

$$MAE = \frac{\sum_{i=1}^{n}(X_i-Y_i)}{n} \qquad \text{Equation (8)}$$

here $\bar{X} = \frac{\sum_1^n(X_i)}{n}$ an $\bar{Y} = \frac{\sum_1^n(Y_i)}{n}$ which are averages of measured (Y_i) and predicted (X_i) outputs while n is the number of sample data in the database. While to get an optimised ANN model, the value of "R" should be highest, approaching 1, while the values of "MSE" and "MAE" should be lowest.

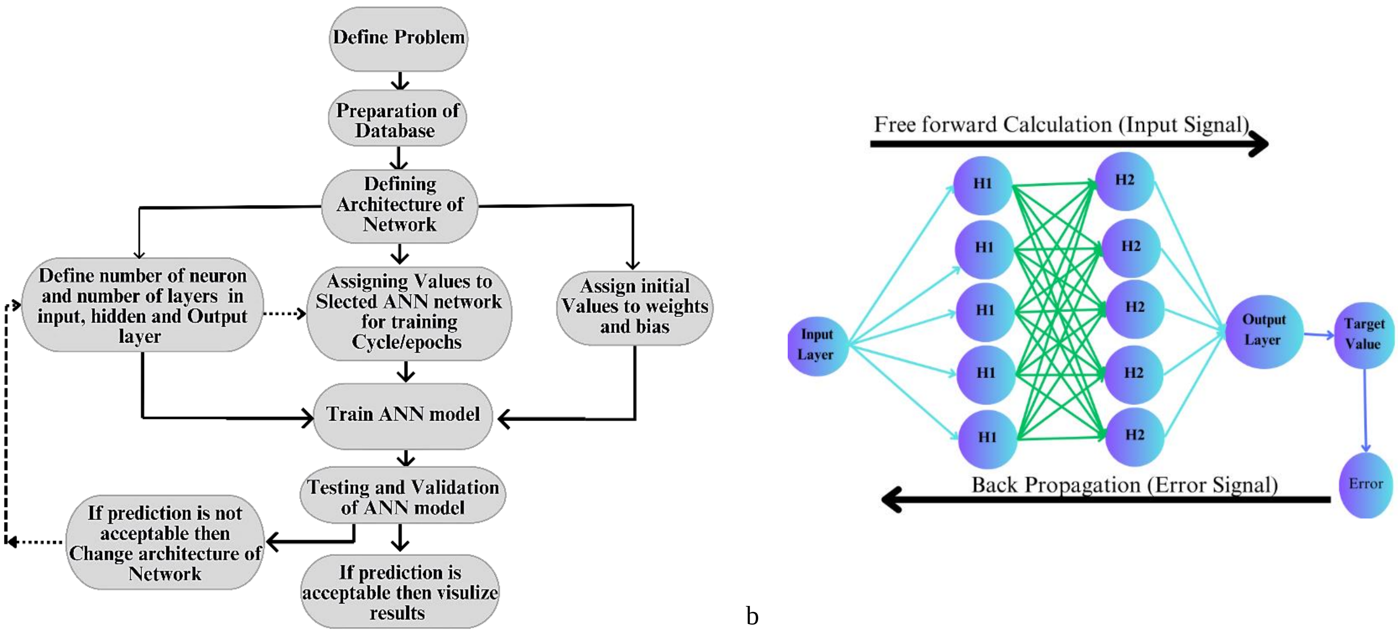
While to overcome the problem of overfitting gradient descent methodology is used to converge values of weights and biases, while at the same time, early stopping criteria as defined in the functionality of ANN is employed to avoid overfitting.

3 Research Methodology

Concrete waste is produced due to research work, participating in global warming to some extent as concrete is also a source of temperature increase. To overcome this on behalf of previous sample data, one of the soft computing techniques, an Artificial neural network, is used to get optimised sample data results to reduce the number of physical concrete beam sample creations. For this research work, the methodology adopted is displayed in the flow chart in Figure 3 (a) to develop a green and economical solution.

4 Results and Discussion

The ANN model's results are compared using regression curves and the coefficient of determination. Regression values range from 0.92 to 0.97, and the coefficient of determination is above 0.94, indicating a good fit and training the model three times yields optimised results. The regression analysis listed in Figure 5 shows precise predictions for the first crack load (R=0.981 for training and 0.967 overall) and flexural strength (R=0.997 for training and 0.977 overall). These results confirm the accuracy of the ANN model in predicting these structural properties.



a b

Figure 3 a) Research Methodology Layout, b) Multilayer Back-propagation Neural Network Structure

Table 2 a) Error checks for the first cracking load model, b) Error checks for the Flexural Strength model

a

First Cracking Load ANN model	
MSE	0.0019
MAE	0.0315
R	0.96735
R^2 (Coefficient of determination)	0.9404

b

Flexural Strength ANN model	
MSE	0.0017441
MAE	0.0234
R	0.97764
R^2 (Coefficient of determination)	0.9756

4.1 Behaviour of First Crack Load based on a/d.

The first cracking load decreases as the shear span to adequate depth (a/d) ratio increases, while keeping the reinforcement ratio constant at 0.349%. Experimental results (Figure 5a) show that for an "a/d" ratio of 1, the load is 28.1 Kip, and for an "a/d" ratio of 6, the load reduces to 2.16 kip. This decreasing trend is due to the increased span length with a higher "a/d" ratio while keeping the depth constant. Comparisons between experimental, ANN, and theoretical results confirm the decreasing trend. ANN predictions achieve a coefficient of determination of 0.9404, indicating accurate predictions based on input parameters.

4.2 Behaviour of first crack load based on reinforcement ratio.

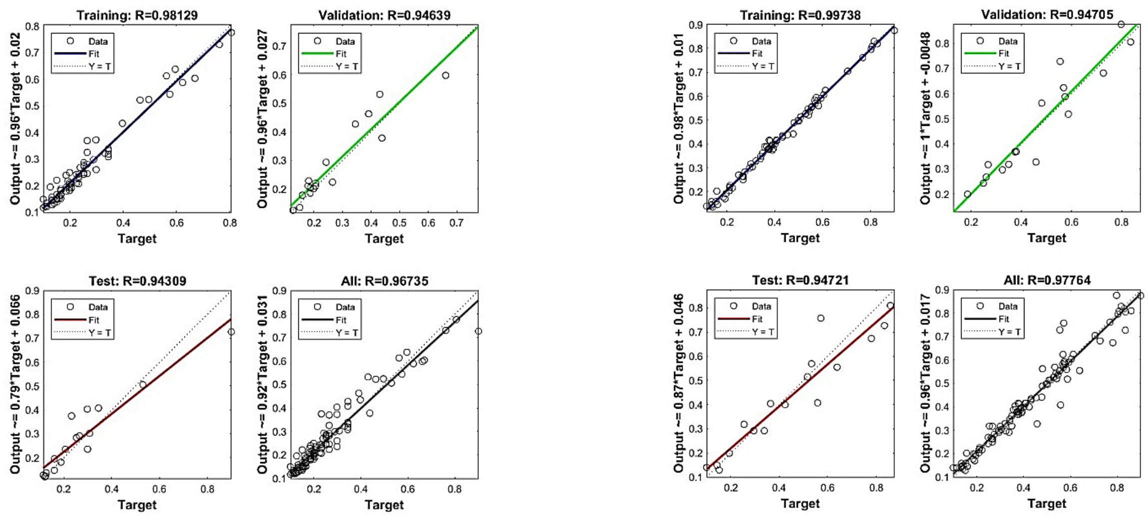
The first cracking load initially increases with a higher reinforcement ratio (%) for a constant shear span to an effective depth ratio (a/d=1) (Figure 5b). However, after reaching a reinforcement ratio of 0.984%, the load decreases for the beam samples. This trend differs from the decreasing trends observed in the same sample's experimental, ANN, and theoretical results. The increase in the first cracking load with a higher reinforcement ratio is due to the steel strength and bond with concrete, enhancing the load-carrying capacity. Excessive reinforcement, however, leads to brittle failure and a decrease in load-carrying capacity.

4.3 Behaviour of Flexural Strength based on a/d.

The Flexural Strength of the reinforced concrete beam with constant steel reinforcement ratio (ρ) 0.349% and varying shear span to effective depth ratio (a/d) shows a decreasing trend with an increase in a/d. The same trend was observed in the case of ANN-predicted results. The prediction accuracy of ANN results is checked by finding the coefficient of determination by comparing predicted results against specific input data with experimental results of that input data physical model that is 0.9756.

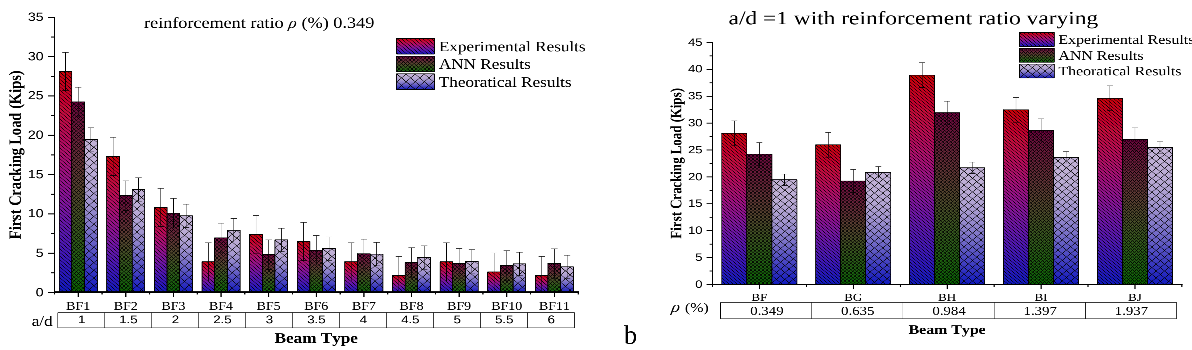
4.4 Behaviour of Flexural Strength based on reinforcement ratio.

Experimental results with a constant shear span to effective depth ratio (a/d) demonstrate an increase in flexural strength up to a reinforcement ratio of 0.984%. Beyond this point, the flexural strength shows marginal changes. Similarly, the ANN-predicted results exhibit the same trend, closely matching the experimental data. This suggests that the ANN model is effective for predicting the flexural strength or ultimate load of reinforced concrete members.



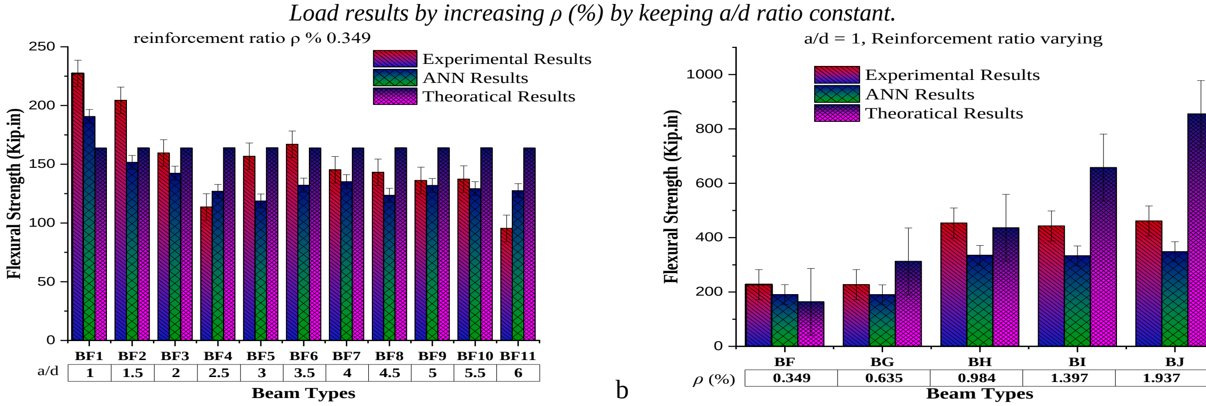
a b

Figure 4 a) ANN model results for First crack load prediction, b) ANN model Results for Flexural Strength prediction



a b

Figure 5 a) Comparison of First Cracking Load results with increasing a/d ratio by ρ (%) constant, b) Comparison of First Cracking Load results by increasing ρ (%) by keeping a/d ratio constant.



a b

Figure 6 a) Comparison of Flexural Strength results with increasing a/d ratio by keeping ρ (%) constant, b) Comparison of Flexural Strength results by increasing ρ (%) by keeping a/d ratio constant.

5 Conclusions

This study explains the development of a knowledge-based structural analysis model capable of predicting RC structural responses. The ANN model was developed based on the Multilayer Backpropagation neural network methodology, which enables the prediction of the behaviour of nonlinear elements. Input parameters are selected based on physical model preparation, and the predicted results are compared with the experimental results of the physical model. Predicted results are close to the experimental results.

1. The linear correlation coefficient "R" for the first cracking load is 0.96735, and the coefficient of determination "R^2" for the First Cracking Load is 0.9404, close to 1, showing precise and close to experimental results.
2. The linear correlation coefficient "R" for the Flexural Strength is 0.97764, and the coefficient of determination "R^2" for the Flexural Strength is 0.9756, close to 1, showing precise and close to experimental results

Unlike the conventional method for predicting the structural response of reinforced concrete members, Soft Computing Techniques, i.e. ANN model, can predict the behaviour of RC structural members with simple or complex geometry under different loading conditions. As soon as the ANN model is trained based on the concerned sample database can predict accurate prediction of the RC structural member accurately without concerning material behaviour and mechanics of the underlying member at the ultimate limit state (ULS).

5.1 Future Wwork Direction

ANN model can be used with professional design software for nonlinear analysis to predict the response of structural members at ultimate limit state under simple or complex loading without requiring analysis time and predicting accurate results compared to conventional current design codes.

Acknowledgement

I wish to express my gratitude to my research supervisor, Prof. Dr. Ayub Elahi for his ample guidance and encouragement to me during this research proceedings.

References

[1] B. S. Institution, *Eurocode 2: Design of concrete structures: Part 1-1: General rules and rules for buildings*. British Standards Institution, 2004.
[2] A. C. Institute, "Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary: An ACI Standard," American Concrete Institute, 2008.
[3] K. Maruyama, "JSCE Standard Specifications for Concrete Structures 2007, Codes in Structural Engineering-Developments and Need for International Practice," in *Joint IABSE-fib Conference Dubrovnik 2010*, 2010, pp. 315–322.
[4] A. Ahmad, N. D. Lagaros, and D. M. Cotsovos, "Neural Network-Based Prediction: The Case of Reinforced Concrete Members under Simple and Complex Loading," *Applied Sciences*, vol. 11, no. 11, p. 4975, 2021.
[5] V. Chandwani, V. Agrawal, and R. Nagar, "Applications of soft computing in civil engineering: a review," *Int J Comput Appl*, vol. 81, no. 10, 2013.
[6] S. K. Das, A. Kumar, B. Das, and A. P. Burnwal, "On soft computing techniques in various areas," *Comput. Sci. Inf. Technol*, vol. 3, no. 59, p. 166, 2013.
[7] S. Talatahari, V. P. Singh, A. H. Alavi, and F. Kang, "Soft computing methods in civil engineering," vol. 2015. Hindawi, 2015.
[8] R. R. Yager and L. A. Zadeh, *Fuzzy sets, neural networks and soft computing*. John Wiley & Sons, Inc., 1994.
[9] B. Z. Dehkordi, R. Abdipour, S. Motaghed, A. K. Charkh, H. Sina, and M. S. Shahid Zad, "Reinforced concrete frame failure prediction using neural network algorithm," *Journal of Applied Sciences*, vol. 12, no. 5, pp. 498–501, 2012.
[10] B. S. En, "1-1. Eurocode 2: Design of concrete structures–Part 1-1: General rules and rules for buildings," *European Committee for Standardization (CEN)*, 2004.
[11] M. D. Kotsovos, "Compressive force-path method," *Switzerland: Springer*, 2013.
[12] Y. O. Özkılıç, C. Aksoylu, and M. H. Arslan, "Experimental and numerical investigations of steel fiber reinforced concrete dapped-end purlins," *Journal of Building Engineering*, vol. 36, p. 102119, 2021, doi: https://doi.org/10.1016/j.jobe.2020.102119.
[13] G. Balaji and R. Vetturayasudharsanan, "Experimental investigation on flexural behaviour of RC hollow beams," *Mater Today Proc*, vol. 21, pp. 351–356, 2020, doi: https://doi.org/10.1016/j.matpr.2019.05.461.
[14] M. H. Beale, M. T. Hagan, and H. B. Demuth, "Neural network toolbox," *User's Guide, MathWorks*, vol. 2, pp. 77–81, 2010.
[15] H. N. Koivo, "Neural networks: Basics using matlab neural network toolbox," *Author Website*, 2008.
[16] A. Ahmad, G. Kotsovou, D. M. Cotsovos, and N. D. Lagaros, "Assessing the accuracy of RC design code predictions through the use of artificial neural networks," *International Journal of Advanced Structural Engineering*, vol. 10, pp. 349–365, 2018.

[17] D. Anderson and G. McNeill, "Artificial neural networks technology," *Kaman Sciences Corporation*, vol. 258, no. 6, pp. 1–83, 1992.

[18] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *J Microbiol Methods*, vol. 43, no. 1, pp. 3–31, 2000.

[19] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, no. 1, pp. 43–62, 1997.

[20] A. Ahmad, D. M. Cotsovos, and N. D. Lagaros, "Framework for the development of artificial neural networks for predicting the load carrying capacity of RC members," *SN Appl Sci*, vol. 2, pp. 1–21, 2020.

[21] A. Ahmad, M. Elchalakani, N. Elmesalami, A. El Refai, and F. Abed, "Reliability analysis of strength models for short-concrete columns under concentric loading with FRP rebars through Artificial Neural Network," *Journal of Building Engineering*, vol. 42, p. 102497, 2021, doi: https://doi.org/10.1016/j.jobe.2021.102497.

[22] M. Kaczmarek and A. Szymańska, "Application of artificial neural networks to predict the deflections of reinforced concrete beams," *Studia Geotechnica et Mechanica*, vol. 38, Jul. 2016, doi: 10.1515/sgem-2016-0017.

[23] U. Naik and S. Kute, "Span-to-depth ratio effect on shear strength of steel fiber-reinforced high-strength concrete deep beams using ANN model," *International Journal of Advanced Structural Engineering*, vol. 5, no. 1, p. 29, 2013, doi: 10.1186/2008-6695-5-29.

[24] F. Giordano, M. La Rocca, and C. Perna, "Input variable selection in neural network models," *Communications in Statistics-Theory and Methods*, vol. 43, no. 4, pp. 735–750, 2014.

[25] M. H. Beale, M. T. Hagan, and H. B. Demuth, "Neural Network Toolbox™, User's Guide, MATLAB® R2015a, The MathWorks," *Inc., Natick, MA, USA*, vol. 410, 2015.

[26] J. Utans, J. Moody, S. Rehfuss, and H. Siegelmann, "Input variable selection for neural networks: Application to predicting the US business cycle," in *Proceedings of 1995 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, IEEE, 1995, pp. 118–122.

[27] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," *Adv Neural Inf Process Syst*, vol. 7, 1994.