كلية الهندسة بشبرا
FACULTY OF ENGINEERING AT SHOUBRA
ERJ
PRINT ISSN 3009-6049
ONLINE ISSN 3009-6022

ENGINEERING RESEARCH JOURNAL (ERJ)
Volume (54),Issue (2)
April 2025, pp:123-129
https://erjsh.journals.ekb.eg

Utilizing AI Approach for Predicting the Contribution of Lateral Confinement to Enhancement of Reinforced Concrete Columns Compressive Strength

Islam M. Alqalyubi [1,*] , Mohammed Hisham [1], Gehan A. Hamdy [1]

[1] *Department of Civil Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt.*
**Corresponding author*

E-mail address: eslam.alqaliouby@feng.bu.edu.eg, mohamed.abdelkreem@feng.bu.edu.eg, gehan.hamdy@feng.bu.edu.eg

Abstract: This study utilizes artificial intelligence to predict the optimum lateral confinement ratio for concrete columns through an innovative dual-phase approach. First, a comprehensive artificial neural network (ANN) model was developed using analytically generated data encompassing diverse parameters: section geometry, dimensions, concrete unconfined compressive strength, transverse steel yield strength, longitudinal reinforcement ratio, vertical spacing between ties and confinement configurations. Secondly, the model was validated against actual experimental results from the literature, achieving its ability to accurately predict the confinement effectiveness coefficient with a mean square error of 0.01 and a mean absolute error of 0.07. This methodology bridges the gap between theoretical models and experimental findings while providing a practical tool for estimating confinement effects without extensive laboratory testing or complicated mathematical formulas.

Keywords: Artificial neural network (ANN), confined concrete, confined concrete strength, compressive strength, concrete column.

Nomenclature:

b	*Side length of square section or diameter of circular section, mm*
f_c'	*Unconfined compressive strength, mm*
s	*Vertical spacing between the centerlines of transverse reinforcement, mm*
f_{cc}	*Confined compressive strength, MPa*
f_{yh}	*Yield strength of transverse steel, MPa*
ρ	*Ratio of longitudinal reinforcement area to the gross area of the concrete section, %*
Φ_{tie}	*Dimension of the horizontal rebar, mm*

1. Introduction

Many researchers have investigated the impact of reinforced concrete (RC) column confinement on enhancing their compressive strength and ductility [1], [2], [3], [4], [5], [6]. Their findings indicate that various parameters contribute to this enhancement, including section dimensions, section shape, horizontal steel confinement configuration, and longitudinal reinforcement ratio. However, with the large number of important parameters and the wide range of values for each parameter, experimental studies typically focus on a limited subset of these parameters within specific ranges.

Based on their experimental studies, many researchers have established mathematical formulas that generally produce accurate results when applied to the specific experimental data from which they were derived. However, the accuracy tends to decrease when these formulas are used with data from other experimental studies, as the conditions and parameters may vary. Establishing an experimental program to study the effect of each effective parameter is complex and expensive, as it requires testing many specimens. Additionally, deriving a mathematical formula from such an extensive experimental study that accounts for this large number of variables is challenging from a mathematical perspective.

As an alternative to conducting such expensive experimental programs, many researchers have employed artificial intelligence (AI) approaches to predict the effects of confinement. One of the most commonly used AI techniques is the Artificial Neural Network (ANN), a highly efficient machine learning algorithm capable of handling classification and regression problems. The availability of enough data for training and validation is the main need for creating an accurate ANN model.

Review of literature related to concrete confinement enhancement reveals limitations in applying Artificial Neural Networks (ANNs) to evaluate concrete confinement effects. Generally, the main issue was the relatively small datasets used for model development compared to the number of input parameters. For instance, Oreta and Kawashima [7] developed an ANN model with 7 input parameters using only 29 training and 9 testing samples for circular concrete columns. Similarly, Ertekin Öztekin's [8] studied square concrete columns confinement utilizing data from 16 experimental studies (232 training and 25 testing samples), which still maintained a modest dataset relative to the model's complexity and parameter space. To overcome this limitation in experimental data availability and its impact on model accuracy and performance, this research proposes an innovative approach: utilizing previously published verified strength confinement equations to generate a comprehensive synthetic dataset, thereby enabling the development of a more robust and reliable ANN model.

2. Methodology

The development of an Artificial Neural Network (ANN) model follows several crucial steps. The process begins with data collection and feature selection, followed by data preprocessing and normalization, continues with network architecture design and training and concludes with model evaluation and validation. This section begins with a brief overview of ANN fundamentals, followed by a detailed explanation of each methodological step, including database development, data normalization, and the determination of optimal network architecture.

2.1 Artificial Neural Network Essentials

Artificial neural networks (ANNs), which take inspiration from biological neural networks, simulate the structure and operations of the human brain and nervous system [7]. Even in situations where complex correlations between several variables are unclear, ANNs can analyze them efficiently. ANNs can estimate the end result due to their learning, classification, and generalization processes [9].

The fundamental building unit of an artificial neural network (ANN) is the "neuron" These neurons are organized into layers, with each neuron in one layer connected to one or more neurons in the following layer, as shown in Figure 1. An ANN consists of at least two layers: an input layer and an output layer, with one or more hidden layers in between. The input layer contains neurons corresponding to the number of features or parameters that affect a certain process; 8 parameters were utilized in this research; therefore, the input layer comprises 8 neurons. The output layer represents the result, which includes the confined strength as a ratio of the unconfined strength. The number and architecture of the hidden layers are determined during the training process. As illustrated in Figure 2, each neuron receives an input, performs a mathematical operation and produces an output for the next neuron Figure 2. Generally, the neuron's input in subsequent layers comes from the outputs of the connected neurons in the preceding layer. Each input value is multiplied by a weight assigned to the link between the neurons. These products are summed, and a bias value is added. The resulting summation is then passed through an activation function, which models the interrelationships between the various hidden layers, as expressed by equation 1.

$$O = f(\sum x_i w_i + b) \quad (1)$$

In Equation 1, 'O' designates the output given by the ANNs, 'x' is the input value, 'w' is the coefficient of weight, and 'b' depicts the added bias value.

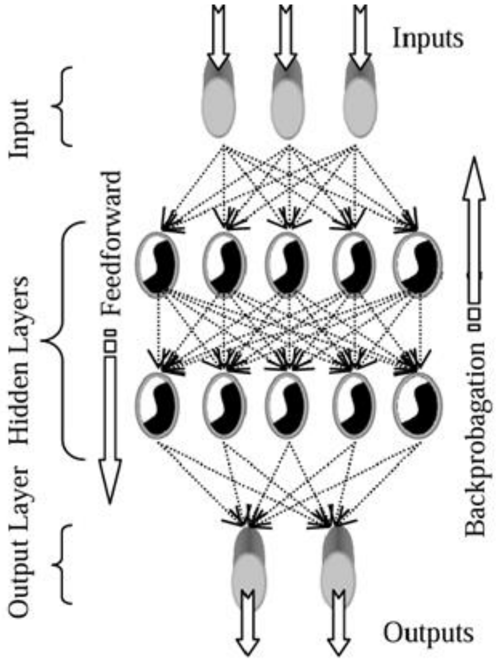

Figure 1: Multi-layer feedforward neural network with backpropagation

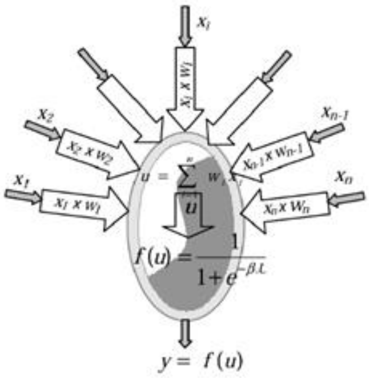

Figure 2: Neuron of an artificial neural network

The training process of an artificial neural network ANN begins with assuming an architecture, which involves determining the number of input neurons equal to the number of features, hidden layers, neurons per hidden layer, and output neurons based on the desired output. Initial assumed weights and biases are assigned for each connection between neurons with small random values. During forward propagation, input data is passed through the network, layer by layer, to produce predictions. The network's predicted output is compared to the actual target values using a loss function, which measures prediction error. To improve accuracy, the error is propagated back through the network in a process called backpropagation, where the gradient of the loss function for each weight is computed. These gradients are used by an optimization algorithm, such as stochastic gradient descent (SGD) or Adam, to update the weights and biases, reducing the loss. This process of forward propagation, loss calculation, backpropagation, and weight adjustment is repeated for multiple iterations, known as epochs, until the loss is minimal. Throughout the training, the network's performance is validated using a separate validation dataset to monitor overfitting or underfitting, and hyperparameters may be tuned accordingly. Finally, the trained model is evaluated on a test dataset to validate its generalization capability, ensuring prediction accuracy for new, unused data.

2.2 Development of Database

The magnitude of the dataset employed for training an Artificial Neural Network (ANN) profoundly influences its capacity to produce precise outcomes. Extensive datasets typically enhance model performance, enabling the artificial neural network to acquire more representative characteristics of the underlying data distribution. This enhances the model's capacity to generalize data by mitigating the impact of overfitting. Limited published experimental studies for concrete column confinement are insufficient to construct a highly accurate model. Therefore, instead of relying solely on experimental samples, this study implemented strength equations derived from previously published works. This approach provides a larger dataset for the confined compressive strength specimens related to different column sections with various dimensions and reinforcement configurations. The data generated was approximately 2,000 column specimens. These data are generated from five published mathematical prediction equations. The parameters considered from each equation and the range of values for each parameter are listed in Table 1. Eventually, these data were used to train and validate the ANN model. Statistical information for the numeric parameters of the generated data is given in Table 2.

2.3 Normalization of Data

Data normalization plays a crucial role in neural network training by ensuring faster convergence and improved model accuracy, as it prevents features with larger scales from dominating the learning process and helps avoid the vanishing gradient problem during backpropagation [10].

In the present study, six numerical features were considered: section dimension, unconfined compressive strength, ratio of vertical reinforcement, tie size, yield strength of tie, and vertical spacing between tie sets. These parameters are reformed into z-score normalization, which standardizes features by transforming them to have zero mean and unit variance [11]. The section shape and confinement configuration parameters represent discrete classes or categories rather than continuous numerical values. Given that neural networks operate exclusively on numerical inputs, appropriate encoding strategies were implemented. For section shape, a one-hot encoding approach was employed, representing the variable as a binary vector of dimension two [12]. For the confinement configurations, an embedding layer [13] was implemented instead of a one-hot encoding method. This approach avoided the need for a seven-dimensional vector representation that could have led to overfitting. The embedding technique enabled a more efficient dimensional representation while maintaining the inherent relationships between configuration categories.

Table 1: Range of parameters studied in adopted analytical research

Study	Concrete Parameters			Reinforcement Parameters				
	Section Shape [a]	Section dimension [b] (mm)	Unconfined compressive strength (Mpa)	Longitudinal reinforcement ratio (%)	Transverse rebar size (mm)	Transverse reinforcement yield strength (Mpa)	Vertical spacing of ties (mm)	Confinement configuration [c]
Sheikh et al. [1]	S	305	[31, 41]	[1.70, 3.70]	[3.2, 10]	[255, 589]	[25, 100]	B, C, D, E
Mander et al. [2]	S, C	500	[24, 32]	[1.20, 3.70]	[10, 16]	[307, 340]	[35, 120]	G
Razvi [4]	S, C	250	[50, 105]	[1.30, 3.90]	[6, 12]	[400, 1000]	[40, 135]	A, B, C, F, G
Bing et al. [5]	S, C	240	[60, 72]	[0.80, 1.60]	6	445	[20, 65]	A, B, G
Yong et al. [3]	S	152	[83.5, 93.5]	1.36, 2.72	3.2	496	[25.4, 152]	B

a. 'S' stands for square sections, while 'C' stands for circular sections.
b. For square sections, the dimension indicates the side length, while for circular sections, it indicates the diameter.
c. Schematic representations of different transverse reinforcement configurations.

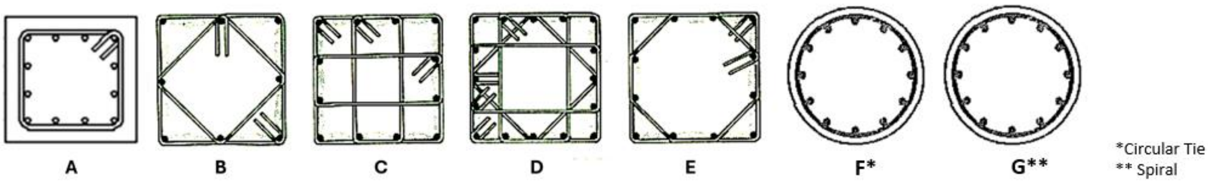

Table 2: The statistical details of the database

Parameter	Unit	Min.	Max.	Avg.	St. Deviation
b	mm	300	600	413.2	109.6
f_c'	MPa	25	85	50.8	21.0
ρ	%	1	3.6	2.4	0.8
Φ_{tie}	mm	8	12	10.0	1.6
f_{yh}	MPa	360	420	390.0	30.0
s	mm	100	200	150.0	50.0

2.4 Determination of Optimum ANN architecture

The developed ANN model utilized a multilayer feed-forward backpropagation (MLFFBP) algorithm [10]. From the 2000 column sections generated, 80% (1600 specimens) were selected for training, while the remaining 20% (400 specimens) were used for validation. The model was then verified using 94 experimental samples obtained from the studies that had developed the strength prediction equations used in this research.

To enhance model generalization and mitigate overfitting, two regularization techniques were implemented. The first technique employed was Dropout, which stochastically deactivates neurons and their corresponding connections within each layer during the training phase, using a predetermined probability p [14]. The second approach incorporated both L1 and L2 regularization methods termed Lasso and Ridge regularization, respectively, which augment the loss function with penalty terms to constrain model complexity [15].

The predictive performance of the model was evaluated using two distinct metrics. The primary metric, Mean Squared Error (MSE), shown in Equation 2, served as the loss function during the training process. The gradient of this loss function guided the iterative optimization of the model parameters across training epochs, to minimize the error. Additionally, Mean Absolute Error (MAE), presented in Equation 3, was employed as a supplementary performance metric to provide comprehensive monitoring of the model's predictive capabilities.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[T_i - O_i]^2 \quad (2)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|T_i - O_i| \quad (3)$$

In Equations 2 and 3, T represents the ground truth (target) values, O denotes the model-predicted outputs generated by the artificial neural network, and n signifies the total number of samples in the dataset.

Through systematic hyperparameter optimization, the optimal ANN architecture was determined. The network consists of a single hidden layer comprising 20 neurons, with a Leaky ReLU activation function. The output layer employs a linear activation function, which is suitable for regression tasks. To mitigate overfitting, dropout regularization with a rate of 0.3 in the hidden layer was implemented, along with L1 and L2 regularization λ1 = 0.003 and λ2 = 0.002, respectively. The confinement configuration feature was represented using a 4-dimensional embedding space. The model was trained using the Adam optimization algorithm for a maximum of 50 epochs, with an early stopping mechanism implemented to prevent overfitting. If no improvement in performance is observed for 10 consecutive epochs, the training is terminated, thereby ensuring computational efficiency while maintaining model performance.

The optimized ANN model demonstrated robust predictive performance across both training and validation sets, as evidenced by the convergence patterns shown in Figures 3 and 4. The model achieved a Mean Squared Error (MSE) of 0.008 on the validation set, while maintaining a comparable MSE of 0.012 on the training set, indicating minimal overfitting. As illustrated in Figure 3, both training and validation MSE curves exhibit smooth convergence, with rapid initial improvement followed by stable performance after approximately epoch 15. The Mean Absolute Error (MAE), which was monitored during training and depicted in Figure 4, reached 0.03 on the validation set, compared to 0.05 on the training set. The close alignment between training and validation metrics (training MAE: 0.051, validation MAE: 0.03) and the parallel convergence patterns observed in both figures further confirm the model's generalization capabilities. These results, along with the consistent downward trends in both error metrics, indicate that the implemented regularization strategies and early stopping mechanism effectively prevented overfitting while maintaining strong predictive performance.

The final trained values of the connection weights and biases for the optimized artificial neural network (ANN) model are presented in Tables 2 and 3. Additionally, Table 4 provides a detailed representation of the trained embedding layer, which is utilized to encode the confinement configuration.

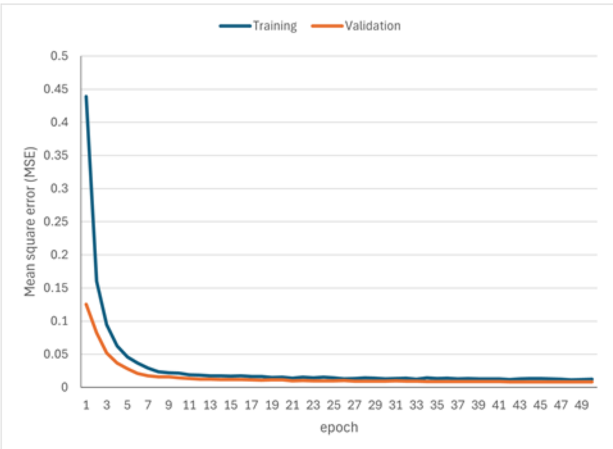

Figure 3: Mean Square Error (MSE) for training and validation sets over epochs

Table 3: Connection weights between normalized input layer neurons and hidden layer neurons, including biases for hidden layer neurons

		Normalized Input Layer Neurons												
		1	2	3	4	5	6	7	8	9	10	11	12	Bias
Hidden Layer Neurons	1	4.39E-05	3.73E-05	3.04E-05	6.43E-05	-1.73E-05	-9.98E-05	-1.03E-04	-7.50E-05	1.93E-04	-4.74E-05	6.31E-05	2.72E-05	-2.26E-01
	2	-2.29E-05	4.01E-05	-3.41E-05	-5.31E-04	-1.34E-04	7.96E-05	3.38E-04	-4.23E-04	-7.24E-05	1.25E-04	-4.06E-05	-1.71E-06	2.47E-02
	3	8.32E-06	2.88E-05	5.06E-05	1.36E-04	-2.03E-04	2.25E-04	-4.78E-05	-2.74E-04	6.84E-05	-2.92E-05	3.87E-05	-7.57E-05	-1.15E-01
	4	-4.50E-05	-9.20E-06	5.24E-05	2.80E-05	2.12E-05	-5.34E-05	2.44E-04	1.48E-04	6.89E-05	-6.23E-05	-1.79E-05	-7.88E-05	1.05E-01
	5	-8.91E-04	5.28E-04	-2.12E-02	-3.96E-02	1.48E-03	2.54E-02	3.99E-03	-9.50E-02	-4.11E-05	5.38E-02	-8.47E-05	-1.34E-04	-5.40E-02
	6	1.40E-04	-4.76E-05	-6.32E-05	6.66E-05	-4.58E-06	-1.08E-05	-2.86E-05	3.88E-05	-7.24E-05	-1.20E-05	6.10E-05	-4.96E-06	-2.42E-01
	7	1.68E-03	-1.47E-04	-2.98E-02	-5.77E-02	2.38E-03	3.98E-02	1.07E-02	-9.48E-02	2.51E-05	5.82E-02	-1.99E-05	-5.96E-05	-6.85E-02
	8	-2.83E-04	-1.82E-04	-1.23E-04	-1.56E-02	6.01E-04	5.41E-03	-6.69E-04	-1.76E-02	1.01E-04	2.23E-03	5.26E-05	1.83E-05	1.91E-02
	9	4.60E-05	6.49E-05	1.15E-04	-8.44E-05	7.50E-05	-1.46E-04	9.98E-06	1.41E-05	-2.73E-05	7.84E-05	-8.34E-05	1.02E-04	1.99E-01
	10	4.22E-05	-2.77E-05	3.68E-05	-1.17E-04	-1.19E-04	-6.45E-05	3.97E-05	-8.66E-05	3.25E-05	1.20E-04	-5.65E-05	-2.26E-05	1.64E-01
	11	-8.91E-05	9.18E-05	6.10E-05	5.32E-07	-5.38E-05	-2.24E-05	2.18E-05	-1.60E-04	-4.66E-06	-2.87E-05	5.44E-05	1.15E-04	-1.31E-01
	12	1.23E-03	2.69E-04	-1.06E-02	-5.67E-02	2.82E-04	2.88E-02	5.12E-03	-6.43E-02	-1.95E-04	4.70E-02	-2.48E-05	7.12E-05	-1.59E-02
	13	-9.55E-07	-1.41E-04	-1.39E-04	1.54E-04	-1.49E-04	-2.11E-04	8.09E-05	1.48E-04	1.72E-04	-1.20E-04	1.41E-05	-2.08E-04	3.39E-02
	14	-5.95E-06	-1.56E-05	1.88E-05	-7.21E-06	5.60E-05	7.44E-05	-3.54E-05	-3.01E-06	-3.75E-06	1.83E-04	4.15E-05	-1.13E-04	-2.23E-01
	15	8.30E-05	-2.35E-05	-9.06E-05	-1.71E-04	5.46E-06	-5.35E-05	-4.52E-06	-7.74E-06	1.48E-04	-4.27E-05	3.15E-05	6.91E-05	-1.94E-01
	16	1.76E-04	1.96E-04	-3.12E-05	1.72E-04	1.25E-04	-3.91E-05	-1.44E-04	-1.81E-04	1.13E-04	5.59E-05	-1.03E-04	3.90E-05	-2.82E-01
	17	-1.04E-03	1.53E-04	-2.16E-02	-5.99E-02	6.78E-04	3.24E-02	2.62E-03	-8.83E-02	5.66E-05	5.31E-02	-3.53E-05	-7.62E-05	-6.53E-02
	18	1.20E-04	1.17E-04	-4.09E-05	5.46E-05	-1.23E-04	1.73E-05	6.67E-05	-6.64E-05	-1.55E-05	1.11E-04	-2.10E-05	-1.36E-05	-2.39E-01
	19	6.76E-04	-5.23E-05	-1.10E-02	-8.35E-02	1.09E-03	3.08E-02	7.35E-03	-4.22E-02	-9.89E-05	5.48E-02	1.01E-04	-6.79E-05	-3.35E-02
	20	-9.70E-04	-1.15E-04	-3.16E-02	-7.34E-02	-1.02E-03	3.62E-02	9.02E-03	-9.49E-02	1.28E-04	6.69E-02	2.44E-04	-2.75E-04	-9.68E-02

Table 4: Connection weights between hidden layer neurons and the output neuron, including the output neuron's bias

		Hidden Layer Neurons																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Bias
Output	1	0.1873	0.1070	0.3255	0.0291	0.5764	0.0848	0.6686	0.2074	0.0109	0.0176	0.2694	0.5171	0.0977	0.1940	0.2778	0.1231	0.6229	0.1650	0.5440	0.7324	1.1479

Table 5: Embedding vectors learned for each confinement configuration (A–G). Each row represents a vector in the embedding space corresponding to a unique categorical configuration.

Confinement Configuration	Dimension 1	Dimension 2	Dimension 3	Dimension 4
A	-0.18277	-1.51719	-0.14402	0.00237
B	-0.02557	-0.02871	-0.14792	0.17486
C	0.11524	0.65162	-0.08318	-0.04109
D	0.28275	1.30981	-0.14464	0.13155
E	0.06467	0.29982	-0.03699	0.13176
F	0.15161	-0.03037	-0.17765	0.13342
G	0.28572	0.04016	-0.24829	0.16776

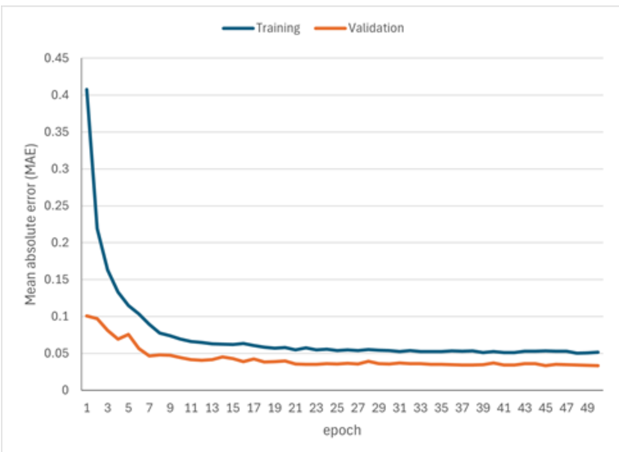

Figure 4: Mean Absolute Error (MAE) for training and validation sets over epochs

3. Results

The performance of the developed ANN model was evaluated by comparing its predictions against experimental measurements of effective confinement [1], [2], [3], [4], [5], as illustrated in Figure 5. The comparison reveals a strong correlation between the ANN predictions and experimental data, with data points generally clustering around the identity line (y=x), indicating good predictive accuracy. The model's performance is quantitatively assessed through error metrics, yielding a Mean Squared Error (MSE) of 0.01 and a Mean Absolute Error (MAE) of 0.07. These relatively low error values confirm the model's capability to capture the underlying patterns in the experimental data. While some data points exhibit deviation from the perfect prediction line, the overall distribution suggests that the ANN model provides reliable predictions of effective confinement.

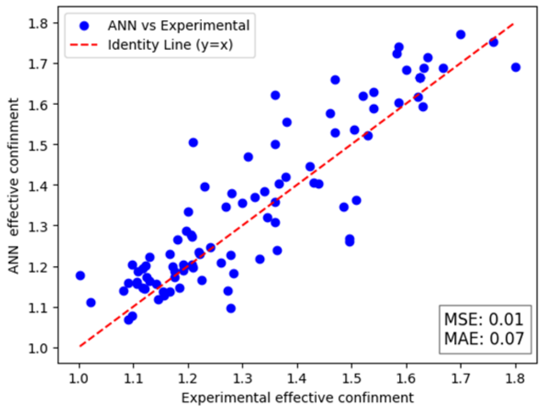

Figure 5: Comparison of ANN-predicted vs. experimental effective confinement values

4. Solved Example

This section demonstrates the calculation of confined compressive strength using two distinct methodologies: an established mathematical formula derived from Bing et al. and the developed Artificial Neural Network (ANN) model. The application utilizes an experimental specimen reported in Mander's study [2], which is a circular cross-section with a diameter of 500 mm, a concrete cover of 25 mm, and a characteristic concrete strength of 29 MPa. The specimen is longitudinally reinforced with 12 - 16 mm rebars and laterally confined using a 12 mm diameter spiral reinforcement with a pitch of 103 mm, with a yield strength of 340 MPa. The experimentally measured confined compressive strength for this specimen was reported to be 40 MPa.

4.1 Mathematical Model Application

The mathematical approach employs Bing's [5] equation for strength calculation, which is demonstrated below:

$$K_e = \frac{\left(1-0.5s'/d_s\right)^2}{1-\rho_{cc}} = 0.91$$

$$f_l' = 0.5 K_e \rho_s f_{yh} = 1.55 \text{ MPa}$$

$$f_{cc} = f_c'\left[-0.413 + 1.413\sqrt{1 + 11.4\frac{f_l'}{f_c'}} - 2\frac{f_l'}{f_c'}\right] = 36.8 \text{ MPa.}$$

4.2 ANN Model Prediction

The developed ANN model provides an alternative computational approach. The ANN implementation is operationalized through software that allows users to input section parameters, eliminating the need for manual calculations. **Error! Reference source not found.** illustrates the computational process occurring within each neuron, utilizing the weights and biases presented in Tables **Error! Reference source not found.Error! Reference source not found.** and applying Equation 1 for neuron output determination.

The ANN model predicts a confined strength ratio of 1.42, corresponding to a confined compressive strength of 41.18 MPa. This result demonstrates improved accuracy compared to the mathematical model prediction of 36.8 MPa when evaluated against the experimental measurement of 40 MPa.

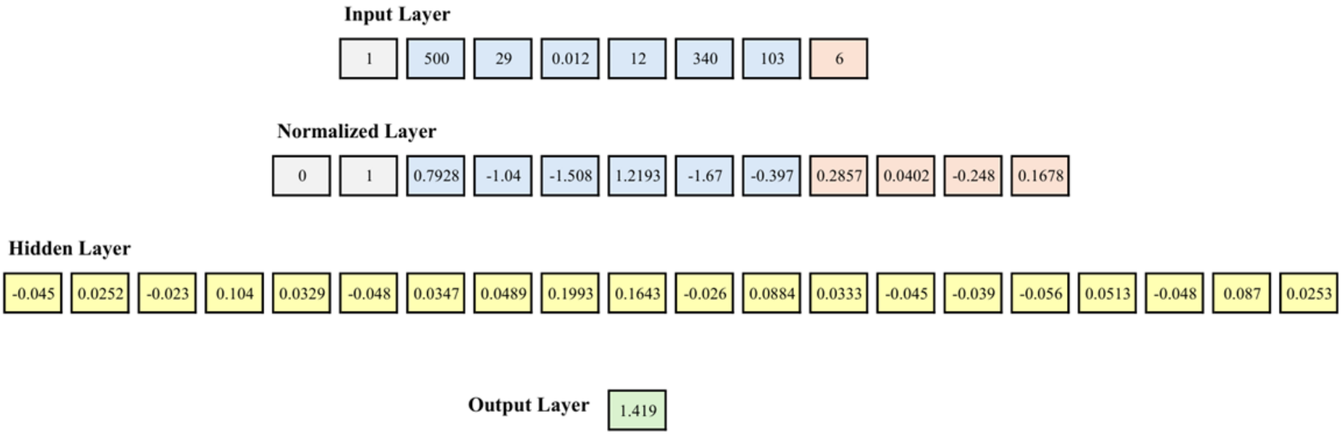

Figure 6: Neural network computation flowchart

5. CONCLUSION

This study presented the application of an artificial neural network (ANN) for predicting the confined compressive strength of square and circular concrete columns. The input variables for the ANN included 11 physical and mechanical properties of confined concrete specimens and 7 different reinforcement configurations. Through extensive trials, the optimal ANN architecture was determined. The performance of the developed ANN model was evaluated by comparing its predictions to experimental results.

1. The study uses artificial intelligence to predict the lateral confinement ratio for concrete columns.
2. An artificial neural network (ANN) model was developed based on diverse parameters such as section geometry, material strengths, and reinforcement confinement details.
3. The ANN model was validated against experimental data, achieving high accuracy with a mean square error of 0.01 and a mean absolute error of 0.07.
4. The approach effectively bridges the gap between theoretical models and experimental results.
5. This method offers a practical tool for estimating confinement effects without requiring complex formulas or laboratory testing.

REFERENCES

[1] S. A. Sheikh and S. M. Uzumeri, "Strength and ductility of tied concrete columns," Journal of the structural division, vol. 106, no. 5, pp. 1079–1102, 1980.

[2] J. B. Mander, M. J. N. Priestley, and R. Park, "Observed stress-strain behavior of confined concrete," Journal of structural engineering, vol. 114, no. 8, pp. 1827–1849, 1988.

[3] Y.-K. Yong, M. G. Nour, and E. G. Nawy, "Behavior of laterally confined high-strength concrete under axial loads," Journal of Structural Engineering, vol. 114, no. 2, pp. 332–351, 1988.

[4] S. R. Razvi, "Confinement of normal and high-strength concrete columns," 1997.

[5] B. Li, R. Park, and H. Tanaka, "Stress-strain behavior of high-strength concrete confined by ultra-high-and normal-strength transverse reinforcements," 2001.

[6] A. M. Abd El Fattah, Behavior of concrete columns under various confinement effects. Kansas State University, 2012.

[7] A. W. C. Oreta and K. Kawashima, "Neural network modeling of confined compressive strength and strain of circular concrete columns," Journal of Structural Engineering, vol. 129, no. 4, pp. 554–561, 2003.

[8] E. Öztekin, "Prediction of confined compressive strength of square concrete columns by Artificial Neural Networks," International Journal of Engineering and Applied Sciences, vol. 4, no. 3, pp. 17–35, 2012.

[9] G. Di Franco and M. Santurro, "Machine learning, artificial neural networks and social research," Qual Quant, vol. 55, no. 3, pp. 1007–1025, 2021.

[10] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in Neural networks: Tricks of the trade, Springer, 2002, pp. 9–50.

[11] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, "Data mining: A preprocessing engine," Journal of Computer Science, vol. 2, no. 9, pp. 735–739, 2006.

[12] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," Int J Comput Appl, vol. 175, no. 4, pp. 7–9, 2017.

[13] C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," arXiv preprint arXiv:1604.06737, 2016.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] A. Y. Ng, "Feature selection, L 1 vs. L 2 regularization, and rotational invariance," in Proceedings of the twenty-first international conference on Machine learning, 2004, p. 78.