Numerical Methods in Civil Engineering, 10-3 (2026) 42-60

Numerical Methods in Civil Engineering

Journal Homepage: https://nmce.kntu.ac.ir/

PSO-Optimized Levenberg–Marquardt Neural Network for Predicting bond strength between concrete and corroded rebar

Narges Zare Tappeh*, Amir Shirkhani, Seyed Ebrahim Sadat Kholerdi***, Iman Afshoon******

ARTICLE INFO

RESEARCH PAPER

Article history:
Received:
November 2025
Revised:
December 2025
Accepted:
February 2026

Keywords:
Monte Carlo simulation; Particle Swarm Optimization; Prediction; Artificial Neural Network; Regression method; Bond strength; Corroded rebar; Concrete.

Abstract:

Rebar corrosion critically affects the durability of concrete structures, necessitating accurate prediction of bond strength between the concrete and corroded reinforcement. This study presents a novel hybrid approach, combining Monte Carlo simulations for systematic selection of the optimal Levenberg–Marquardt-based Multi-Layer Perceptron (LM-MLP) architecture with Particle Swarm Optimization (PSO) for refining network weights and biases. Using 132 experimental data points, the optimized model achieved a maximum correlation coefficient (R) of 0.959, representing an improvement of up to 3.75%, and reduced the root-mean-square error (RMSE) by up to 21.42% compared to the conventional LM-MLP model. An empirical regression model is also developed for comparison, reaffirming the superior accuracy of the proposed approach. These results demonstrate the model's robustness and effectiveness for rapid and reliable prediction of bond strength under varying corrosion conditions. This hybrid approach not only enhances the accuracy and stability of the model but also provides rapid and reliable predictions under varying corrosion conditions, outperforming classical methods.

1. Introduction

The degradation of the bond between reinforcing steel rebars and concrete owing to corrosion remains a substantial concern in evaluating the durability of reinforced concrete (RC) structures. Previous studies have demonstrated that corrosion damage dramatically alters the behavior of the steel-concrete interface, affecting bond strength, stiffness, and failure mechanisms under different environmental and material conditions.

Due to the increased presence of salts in water affecting bridge decks and columns, reinforced concrete corrosion has intensified in recent decades. Steel corrosion products expand up to seven times their original size within the concrete, generating internal pressure and concrete cover cracking. Cover cracking exposes the reinforcing bars to further corrosion [1]. In general, the most common sources of corrosion in concrete reinforcement are inlets containing Cl^- and/or CO_2 [2]. As corrosion progresses, concrete's tensile capability declines rapidly, resulting in enormous fractures. Lubricating corrosion products develop between the concrete and the reinforcement as a result of the severe corrosion. The strength of the connection gradually deteriorates after this stage [3].

Concrete's age has a direct impact on its strength. According to [4] during the first three days following casting, concrete's age has a substantial impact on binding strength. It's also worth noting that at young ages, bond strength grows more quickly than splitting and compressive strength. Pull-out and splitting failures are the two most common

* *Msc, Department of Civil Engineering, University of Mohaghegh Ardabili, Ardabil, Iran*
** *Corresponding author: Assistant Professor, Department of Civil Engineering, University of Torbat-e Jam, Torbat-e Jam, Iran. Email: shirkhani@tjamcaas.ac.ir*
*** *Assistant Professor, Department of Civil Engineering, University of Torbat-e Jam, Torbat-e Jam, Iran.*
**** *Civil Engineering Department, University of Sistan and Baluchestan, Zahedan, Iran.*

https://doi.org/10.61882/NMCE.2511.1105

types of bond failure. Shearing off of the concrete between steel rib pairs is the most common cause of pull-out failure [5, 6]. Splitting failure results from the concrete surrounding the bar splitting longitudinally [7, 8]. When radial fissures reach the element's outer surface, bond capacity drops dramatically. When it comes to the behavior of the bond stress-slip of rebars in concrete, Gambarova and Rosati [9] and Luccioni et al. [10] have discovered that the post-peak behavior is influenced by the concrete quality, particularly the propagation of fractures. The mean bond strength corresponds to the peak stress.

Many research works on the bond strength of corroded reinforcement have been conducted in recent decades [11-16]. Saifullah and Clark [17] looked into how bond strength was affected by the corrosion rate of corroded reinforcement. They discovered that when the corrosion rate exceeded 250 A/cm2, the bonding resistance deteriorated even more. Almusallam et al. [18] investigated the bonding resistance of corroded reinforcement and concrete. According to their findings, the corrosion levels lowered bond strength by 6%. Fang et al. [19] studied the bond strength of corroded rebar and concrete. They claimed that even if corrosion increased by more than 5%, bond strength increased.

Many catastrophic collapses of structures, such as the fall of the Berlin Congress Hall, have been caused by corrosion of reinforcing steel, resulting in injury and death [20]. The effects of steel corrosion on concrete have been extensively researched [21, 22]. These models are consistent with the system's qualitative corrosion assumptions. However, long-term forecasts based on each of these models have yet to be reported [23]. Numerical and analytical approaches such as corrosion modeling and forecasting chloride profiles, are necessary to anticipate and understand the behavior of steel in concrete. In this scenario, the artificial neural network (ANN) can be employed for modeling and forecasting. ANNs are interconnected networks made up of many computing neurons (or units) with a distributed parallel processing topology [24].

An input layer, hidden layers, and an output layer make up an ANN. The input layer neurons only function as distributors, while the hidden layer neurons contain weights and perform the operating actions. Consequently, the ANN is trained with the input and output data to predict the results of the next input data. Each network's data can be split into training and testing sets. It's vital to choose the appropriate number of training data. But there are no general guidelines for determining the size of training data. A considerable number of samples display specific characteristics that are required in the training domain [23]. In recent decades, many research works have been performed using neural network modeling on concrete experimental data [6, 25-27]. Tanyildizi [28] recently used deep learning and extreme learning machine models to estimate the bond strength of corroded rebar in concrete. He discovered that the deep learning model is more accurate than the extreme learning machine model at anticipating the bond strength of corroded rebar.

However, in a study a little earlier, Shirkhani et al. [29] utilized the Multi-Layer Perceptron Artificial Neural Network (MLP-ANN), which is one of the most well-known and popular types of neural networks [30-33], to anticipate the bond strength between corroded rebar and concrete. They determined the structure and parameters of the used ANN by experience. Their findings indicated that ANN could be utilized as a beneficial tool to forecast the bond strength between corroded rebar and concrete.

Based on a comprehensive review of the literature, no known study has simultaneously (i) evaluated the performance of different ANN structures for predicting the bond strength of corroded rebar in concrete under input variability using a Monte Carlo–based framework, and (ii) optimized the parameters and weights of the selected network with a metaheuristic algorithm such as Particle Swarm Optimization (PSO). In this study, Monte Carlo simulations are first employed to assess the convergence and robustness of various LM-based Multi-Layer Perceptron (LM-MLP) structures across the training, testing, and validation datasets. Following this, the PSO algorithm is applied to optimize the parameters and weights of the selected optimal LM-MLP architecture. A total of 132 experimental data points is used, and performance is evaluated using standard indicators, including the correlation coefficient (R) and root-mean-square error (RMSE). Finally, an empirical regression-based relationship is developed for comparison with the results acquired from the optimized LM-MLP model.

This study presents a novel approach that integrates Monte Carlo simulations for systematic selection of neural network architectures with Particle Swarm Optimization (PSO) for refining network weights and biases. Unlike prior models that often rely on fixed architectures or heuristic parameter tuning, this framework provides a comprehensive and robust optimization process, enhancing prediction accuracy and generalizability for the bond strength of corroded reinforcing bars. This combination of stochastic architecture evaluation and metaheuristic optimization distinguishes our work by offering improved adaptability and reliability in modeling complex interactions inherent in corrosion-affected bond behavior.

2. Research objective and concepts

The goal of the present study is the optimization of the LM-based MLP neural network's parameters and structure using

the PSO algorithm to anticipate the bond strength of corroded rebar in concrete. Therefore, the current work's contribution is based on the following concepts: (1) a comprehensive and detailed investigation is conducted to choose the best LM-MLP structure to predict the bond strength of corroded rebar in concrete; (2) Statistical analysis and Monte Carlo simulations are employed to assess the LM-MLP structure's performance while considering the effect of random dataset splitting [34]; (3) this is a highly accurate and comprehensive study that attempts to anticipate the bond strength of corroded rebar in concrete under the chloride ion penetration by optimizing the parameters and structure of the ANN; (4) for enhancing the robustness of the optimal LM-MLP structure selected based on Monte Carlo simulations, the forecast accuracy is improved by employing the PSO algorithm; (5) on the basis of the classical regression method, an empirical relationship with high accuracy is also developed and compared with the results acquired from optimized LM-MLP. The overall modeling framework follows the sequential steps summarized below, encompassing data preprocessing, Monte Carlo-based architecture evaluation, Levenberg-Marquardt training, and Particle Swarm Optimization for final model refinement:

1. Load Dataset
2. Preprocess Data:
a. Normalize input and output variables using min-max scaling
b. Define min and max values for each variable from the training subset

3. Define Parameters:
- Number_of_MC_Runs = 1000
- ANN_Structures = list of candidate architectures (e.g., varying hidden layers/neurons)
- PSO Parameters (swarm size, iterations, inertia weight, learning coefficients)

4. For each ANN_structure in ANN_Structures do:
a. Initialize Performance_Metrics_List = empty
b. For run = 1 to Number_of_MC_Runs do:
i. Randomly split data into Training, Validation, and Testing subsets
ii. Initialize ANN weights randomly
iii. Train ANN with Levenberg-Marquardt (LM) algorithm on Training subset
iv. Evaluate ANN performance on Validation and Testing subsets (compute RMSE, R)
v. Store performance metrics in Performance_Metrics_List
c. Compute average and standard deviation of performance metrics across all runs
d. Select optimal ANN_structure based on performance (e.g., lowest average RMSE)

5. Optimize Selected ANN_structure weights and biases using Particle Swarm Optimization (PSO):
a. Initialize swarm population randomly within the feasible weight space
b. For iteration = 1 to Max_Iterations do:
i. Evaluate fitness (e.g., RMSE on Training data) for each particle
ii. Update particles' velocities and positions using PSO update rules
iii. Store the best solutions found so far
c. Finalize optimized weights and biases
6. Evaluate optimized ANN on Testing and Validation datasets
7. Report final performance metrics (RMSE, R) and compare with baseline
8. (Optional) Provide code and data availability statement for reproducibility

3. Materials and methods

3.1. Used dataset

Amleh [20] investigated the bond deterioration of rebar in concrete owing to corrosion in the pullout testing program in 2000, using different concrete mixtures and different thicknesses of concrete cover (25, 50, 75, and 100 mm). The types of concrete mixtures used in the specimens are as follows [20]:
C1: Point Tupper (fly ash w/scm = 0.32)
C2: Thunder Bay (fly ash w/scm =0.32)
C3: Sundance (fly ash w/scm = 0.32)
C4: Normal portland cement (w/c = 0.32)
C5: Normal portland cement (w/c = 0.42)
C6: High aluminum cement (w/c = 0.37)

To limit the extent of the research, the bar diameter, steel strength, and anchorage length have been kept constant as per Almeh's study [20]. For the pullout test, embedded length and specimen length have been variables [20]. In this research, data of 132 concrete samples [20] are analyzed utilizing ANN to anticipate the bond strength between concrete and steel rebar exposed to chloride ion penetration (which causes steel corrosion in concrete). ANN can detect possible interplay between predictors and target variables through iterative regulation. Different training algorithms can be used to robustly derive the characteristics of the predictor data [35, 36]. The statistical information of the data utilized in this research, such as the minimum, maximum, median, mean, standard deviation (StD), and all variables' skewness, are shown in Table 1. In this research, 70% of the data is considered as the training part of the ANN, and 30% of it is used in equal contributions for testing and validation parts. The histograms of input variables (concrete cover thickness, mass loss owing to corrosion, the width of

longitudinal crack owing to corrosion, and chloride ion content at the rebar level) and corresponding correlation analysis with the bond strength between concrete and corroded rebar are displayed in Fig. 1.

All variables comprising output and input variables of the used data are normalized to a scale of zero to 1, according to Eq. (1) [34]:

$$Z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (1)$$

In which x indicates the output or input variable that needs to be normalized, x_i denotes the corresponding value, and z_i indicates the computed normalized value. In the data normalization process, the minimum and maximum parameters for each input and output variable are calculated solely based on the training dataset of each Monte Carlo run, and the same scaling is then applied to the validation and testing sets. This standard approach is employed to prevent data leakage and maintain the independence of the datasets. Specifically, the potential impact of data leakage in this study is minimized due to the relatively small dataset size and the high number of runs conducted.

3.2. Artificial Neural Network (ANN)

To resolve the forecast problems, ANN as a popular computational model comprising multiple processing elements is used [37-40]. The nature of ANN is derived from research into neural biological systems. The ANN comprises interconnected elements and nodes. It can be divided into an input layer, hidden layers, and an output layer.

As an outcome of the training stage, the ANN supplies a model that can forecast a target parameter from a specified input value [41]. The main benefit of the neural network is that it doesn't require an explicit model for explaining the relationship between output and input variables, resulting in a straightforward parametric procedure [42]. In the ANN model, an intricate correlation between output and input variables can be modeled with high accuracy compared to the conventional techniques.

In fact, the correlation among output and input variables in a neural network model is made through the data points themselves. They are extremely independent, allowing multiple operations to be performed in parallel [34].

3.3. Particle swarm optimization (PSO)

Eberhart and Kennedy [43] developed PSO, which is akin to a genetic algorithm. An initial value is given to a population of random solutions in that system. Then, the potential solutions, known as particles, are ‘‘flown’’ via hyperspace by allocating a given velocity to each potential solution [43]. The theoretical framework of PSO is easily understood. It can be coded and executed efficiently and is computationally affordable according to CPU and memory necessities [44]. For nonlinear continuous functions, the PSO algorithm can be used as an optimizer [45]. A particle, or swarm member in the PSO, is a possible solution, or, to put it another way, a point in the search space. The particle has a quickness and a fitness value to alter the flight path to the best swarm experiences in order to discover the global optimum in the space of the D-dimensional solution. Recently, PSO has received a lot of interest and has been effectively applied in different areas, particularly for unconstrained continuous optimization problems [44]. Due to the following reasons, the PSO algorithm is chosen for this research:

First, such an approach is simple to grasp and use, especially in a popular matrix environment such as MATLAB [44, 46]. Second, the PSO algorithm has been extensively utilized in different fields of optimization research. The employment of PSO algorithms in the optimal solution of issues produced better results than traditional methods. PSO has a faster convergence rate than other methods (e.g., genetic algorithm, simulated annealing, differential evolution, etc.) [45] because it utilizes two populations (p_{Best} and g_{Best}) in its algorithm, which allows higher exploration and diversity compared to a single population in other algorithms. Lastly, because PSO is entirely parallelizable, it has a high processing power [47].

Table 1. Statistical information of the data utilized in this research.

	Symbol	Unit	Min	Median	Mean	Max	StD	Skewness
Concrete cover thickness	t_{cc}	mm	25	50	61.74	100	27.53	0.03
Mass loss owing to corrosion	m_l	%	0.13	6.86	7.53	28.48	5.97	1.26
Width of longitudinal crack owing to corrosion	w_c	mm	0	0.8	1.27	6	1.37	1.32
Chloride ion content at rebar level	C_{Chl}	%	0.01	0.165	0.32	1.87	0.37	1.86
Bond strength between concrete and corroded rebar	u	MPa	0.24	5.42	5.59	12.89	2.87	0.17

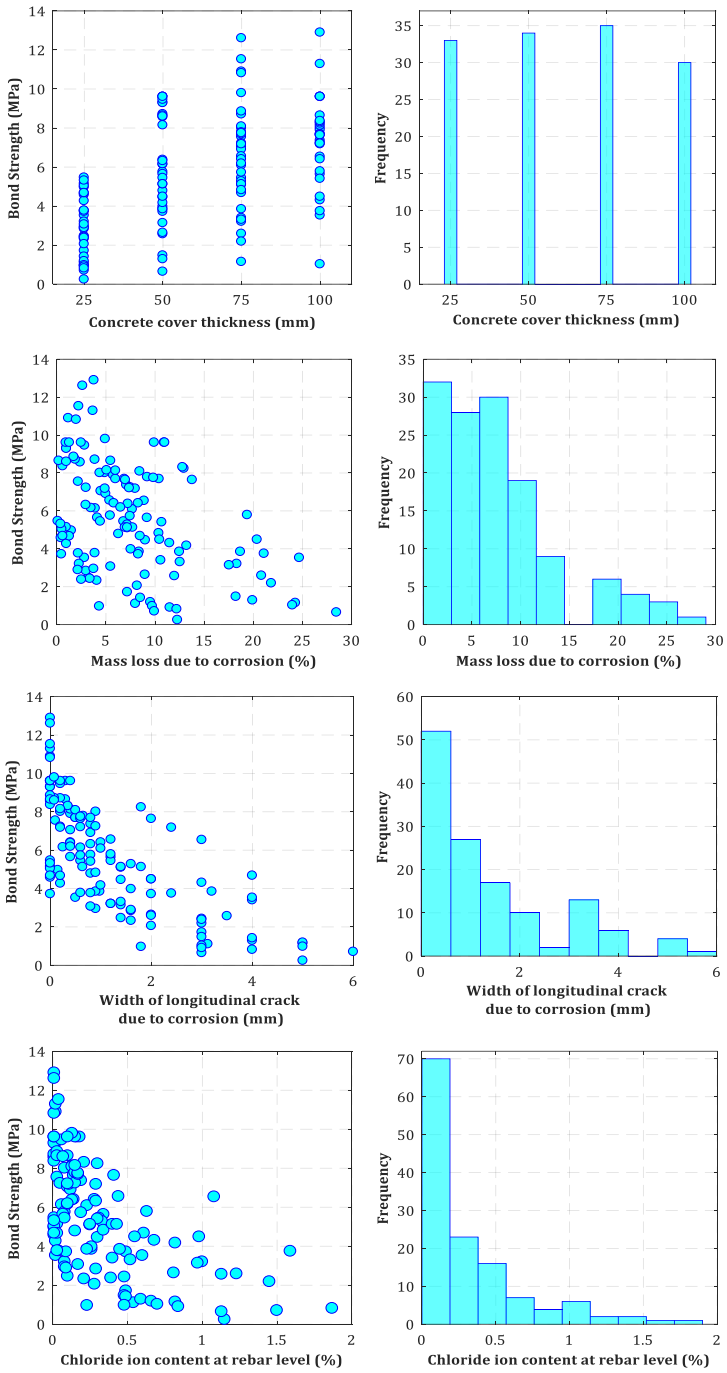


Fig. 1: Correlation analysis between bond strength and used input variables histogram in this research.

3.4. Quality evaluation indicators

The quality evaluation indicators are utilized to assess the model's accuracy in this research, including the correlation coefficient (R) and the root mean square error (RMSE). The formulas of these criteria are as follows [34, 48-52]:

$$R = \sqrt{1 - \frac{\sum_{i=1}^{N}(x_a - x_m)^2}{\sum_{i=1}^{N}(x_a - \bar{x}_i)^2}} \quad (2)$$

$$RMSE = \sqrt{\sum_{i=1}^{N}(x_a - x_m)^2 / N} \quad (3)$$

In which, N denotes the total number of input data, x_a and x_m denote the actual measured values and model values, respectively, $\bar{x}_i$ are the evading index's mean values. The correlation coefficient is valued in the range of $-1 \leq R \leq 1$. A prediction that has an absolute value of R nearly 1 is quite successful.

3.5. Monte Carlo approach

The Monte Carlo approach is employed to have a thorough evaluation of the neural network algorithm performance considering the variability in the input space and also to quantify the models' robustness. The Monte Carlo approach has been generally known in data science, due to its efficiency and the capability to large-scale parallel computing [34]. The model coefficient's variability and the relative weights of the system's parameters can be handled via Monte Carlo simulation, an extensively used procedure for uncertainty-based system design [53-56]. Here, Monte Carlo simulations refer to the procedure of performing as many simulations as feasible using the neural network algorithm for computing the output, considering the random sampling approach. Hence, to evaluate the algorithm's efficiency, a statistical measure has to be considered. Thus, the following equation can be used to define the convergence estimator [34]:

$$Conv.Estim.(n_M) = \frac{1}{R}\frac{1}{n_m}\sum_{j=1}^{n_M} Rj \quad (4)$$

In which, $\underline{R}$ indicates the random variable's average value R under consideration and n_M denotes the number of Monte Carlo runs. The Monte Carlo procedure for each neural network architecture was conducted as follows: initially, the data were randomly split into training (70%), validation (15%), and testing (15%) sets. This data partitioning varied with each Monte Carlo run, with each run involving a new random split. Additionally, the initial weights of the network were randomly initialized. For each architecture, 1000 independent runs were performed, each involving model training using the LM algorithm and performance evaluation on the validation and testing datasets. After completing all runs, the mean and standard deviation of the evaluation metrics, including RMSE and correlation coefficient (R), were computed across the validation and testing sets. This approach allowed for a comprehensive assessment of the model's performance considering the randomness in both data splits and initial weights.

4. Results and discussion

4.1. Model's convergence analysis

The parametric research in this paper is focused on determining the number of hidden layers and the number of neurons in each hidden layer. The activation functions, the training function, and the cost function of the ANN model are all fixed and presented in Table 2. Because the MLP neural network model employs the Levenberg-Marquardt (LM) algorithm, it is referred to as the LM-MLP model in this paper.

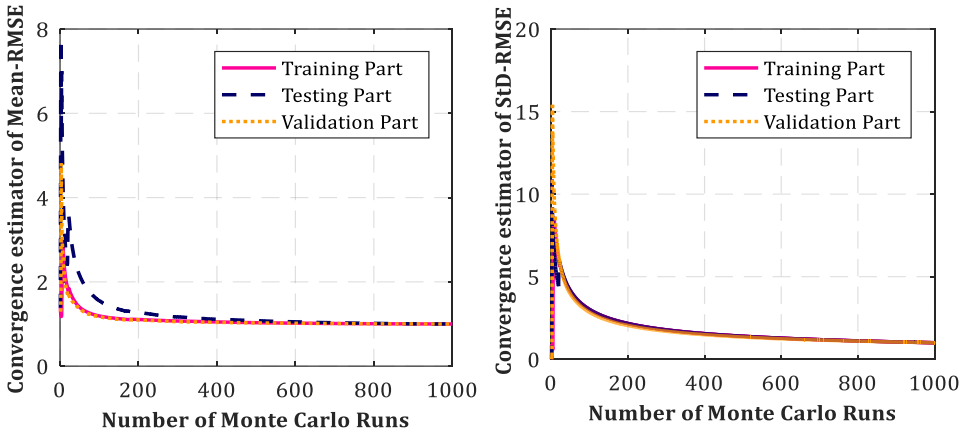
Table 2. Characteristics of ANN in this research.

Parameter	Value
Input layer	Number of neurons =4
Output layer	Number of neurons = 1
Hidden layers	Number of neurons = varying from 1 to 10
	Number of hidden layers = varying from 1 to 2
Hidden layer's activation function	Sigmoid
Output layer's activation function	Linear
Cost function	Mean Square Error (MSE)
Training algorithm	Levenberg-Marquardt (LM)

A total of 110 LM-MLP structures are produced by altering the number of hidden layers and neurons, including 10 structures with a one hidden layer and 100 structures with two hidden layers. For each structure, a total of 1000 Monte Carlo runs are performed to shuffle the samples' indexes that emerged in the training dataset. Figs. 2 and 3 show the typical outcomes of one chosen LM-MLP structure (i.e., LM-MLP-[4-3-3-1]). 1000 Monte Carlo simulations are able to produce reliable convergent results from the training, testing and validation datasets, as can be observed. For several simulations below 200, a slight fluctuation of 1% around the average value (indicated by μ) is noticed regarding the value of RMSE (Fig. 2 (a)). On the other hand, to stabilize around 1% of the mean values for R, about 50 Monte Carlo runs are needed (Fig. 3(a)). 300 simulations are discovered as the proper number regarding the value of the standard deviation (denoted as σ) of RMSE, where the value of σ converged. To stabilize σ_R around the mean values, around 600 simulations are considered. As the number of Monte Carlo simulations increases, the mean and standard deviation of the neural network errors each independently trained with randomized data splits converge towards stable values representing the true population. Although individual errors may vary across runs, the statistical properties of the

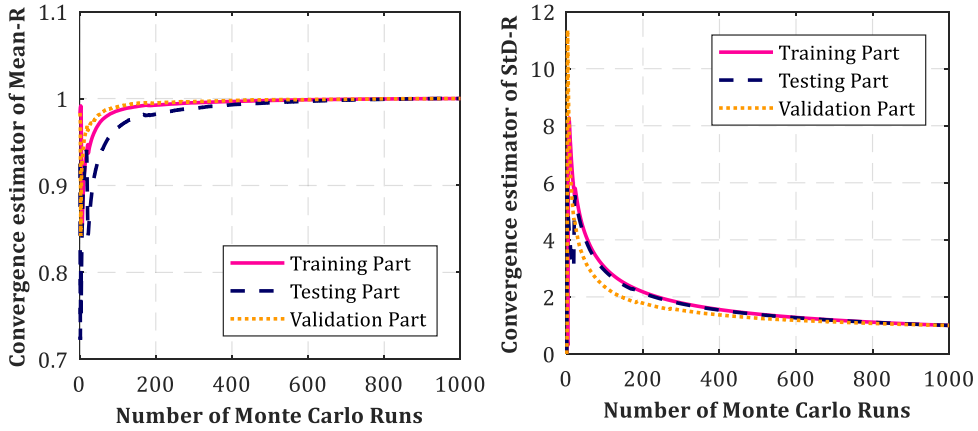
Monte Carlo method reduce estimation fluctuations and improve the accuracy of the model's average performance, resulting in the convergence of error plots and statistical indicators shown in Figs 2 and 3. Considering the least n_M needed to generate converged simulation results, the remaining LM-MLP structures are illustrated in Figs. 4 and 5.

The results' fluctuation within the 1% range of the mean values of RMSE and R for the training, testing, and validation datasets is chosen as the criterion. As can be observed, a number of $n_M = 300$ is required for R, while a number of $n_M = 900$ is adequate for RMSE of the training, testing, and validation datasets. Therefore, it can be said that considering the random sampling effect of data, all 1000 nM results provided by LM-MLP are reliable. Generally, it can be seen that the LM-MLP structure has a good performance, and the results are statistically converged. In the present study, to ensure a comprehensive exploration of the neural network architecture space, architectures with one hidden layer (10 structures) and two hidden layers (100 structures) were evaluated. Although it is widely established in the scientific literature that networks with two hidden layers have superior capability in modeling complex nonlinear relationships, single-layer architectures were systematically examined to compare the performance of simpler models and to avoid unnecessary complexity. The obtained results showed that two-hidden-layer models generally perform better, leading to the selection of the optimal LM-MLP-[4-3-3-1] structure with two hidden layers. This approach facilitated the assessment of architectural diversity and significantly enhanced the validity and robustness of the results.



(a) **(b)**

Fig. 2: RMSE statistical convergence for 1000 random samples: (a) Mean; (b) StD.



(a) **(b)**

Fig. 3: *R* statistical convergence for 1000 random samples: (a) Mean; (b) StD.

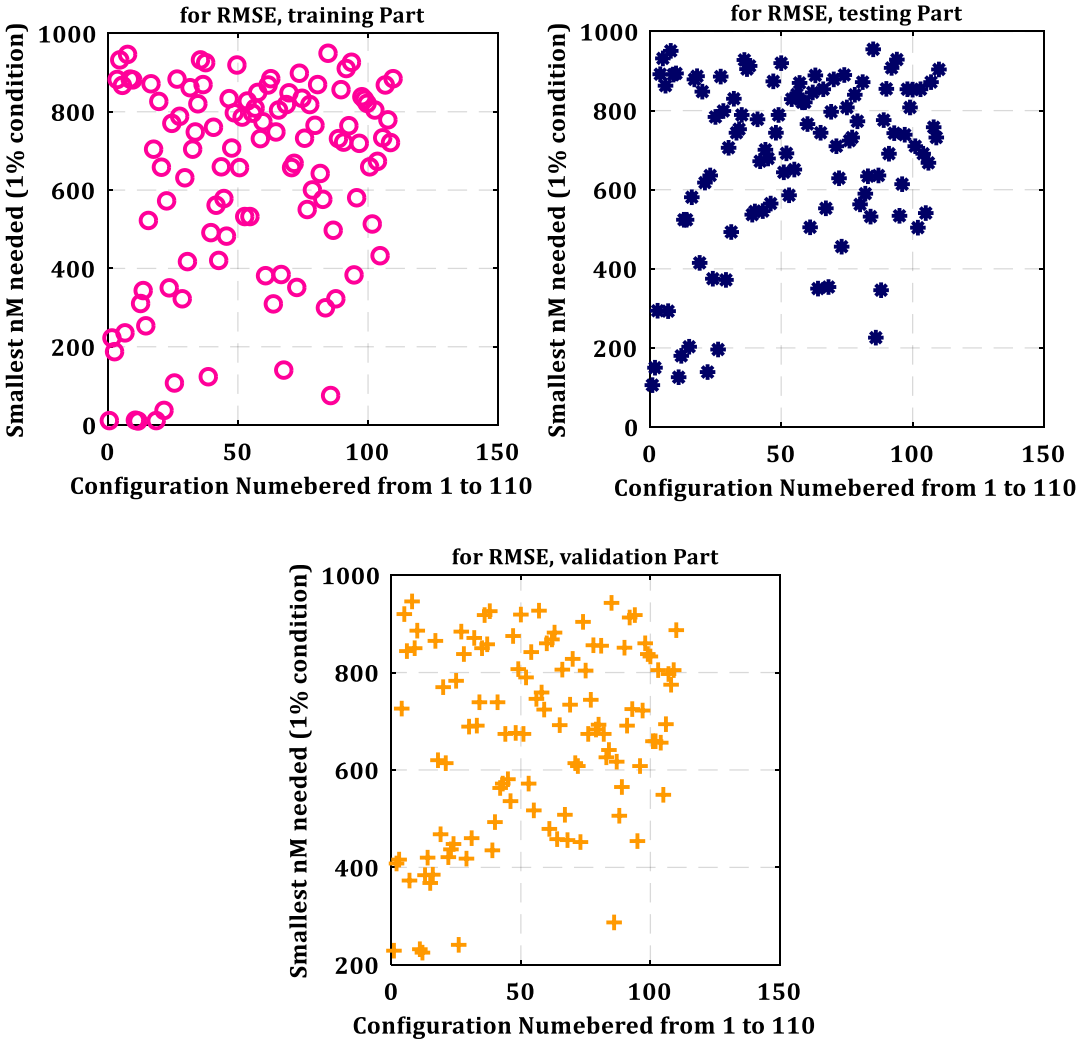


Fig. 4: Least η_M needed for results stabilization within the 1% range of mean values, RMSE for different parts of data used in this study.

4.2. Selecting optimal architecture of LM-MLP

For the training, testing, and validation datasets, the RMSE indicator values of various LM-MLP structures are examined, as shown in Fig. 6. The values of μ_{RMSE} and σ_{RMSE} over 1000 nM for the training, testing, and validation datasets are displayed in Fig. 6. Here, the maximum number of neurons per layer is 10. According to Fig. 6 (a), the best results, which have less error for μ_{RMSE}, are between 6 to 8 neurons in the monolayer state, and between 4 to 10 neurons in both layers in the 2-layers state. Of course, these results are different for σ_{RMSE}. In fact, the results of σ_{RMSE} (Fig. 6 (b)) do not follow a specific pattern. However, Fig. 6 (a) shows an ideal zone in the training, testing, and validation datasets' mean values. The values of μ_R and σ_R over 1000 nM for the training, testing, and validation datasets are shown in Fig. 7. According to Fig. 7 (a), the maximum approximate value provided for μ_R in the training, testing, and validation datasets is the same and is more than 0.95. As can also be seen in Fig. 7 (b), the results of σ_R for the training part don't follow a pattern similar to other parts. Based on the examination of the LM-MLP structure, it can be concluded that: (i) utilizing two hidden layers can give better forecast accuracy than using a single hidden layer, and (ii) increasing the number of neurons in both hidden layers up to the maximum number of neurons in this study guarantees an increase in LM-MLP accuracy. According to Figs. 6 and 7 and considering the results of μ_{RMSE}, σ_{RMSE}, μ_R, and σ_R, there is no common area between these figures. Therefore, 10 structures in each criterion (including μ_{RMSE}, σ_{RMSE}, μ_R, and σ_R) are selected (according to Fig. 8) which have the best results, and choose the structure that has the most repetitions among the results as the optimal structure of the neural network. A structure that has two hidden layers and 3 neurons in each layer existed in all the best common results and is selected as the optimal structure.

Therefore, the LM-MLP-[4-3-3-1] architecture (As shown in Fig. 9) is chosen as the optimal architecture of LM-MLP based on the exact values of μ_{RMSE}, σ_{RMSE}, μ_R, and σ_R over 1000 simulations using 110 LM-MLP structures.

4.3. Optimization of LM-MLP by PSO algorithm

4.3.1. Cost function utilizing PSO

The bias values and weights of the selected LM-MLP structure are optimized using the PSO algorithm. Table 2 lists the characteristics of the LM-MLP model, whereas Table 3 lists the parameters of the PSO optimization approach.

A maximum number of 500 iterations is employed in this research to see if the stopping requirement is met when employing the PSO optimization technique. The PSO algorithm is also performed with a swarm size of 20. Based on *R* and RMSE, Fig. 10 displays the optimization cost for the training, testing, and validation datasets.

The PSO algorithm demonstrates a powerful capability for optimizing the parameters of the primary LM-MLP model, as can be observed. Furthermore, because all cost functions are converged at about 300 iterations, 500 iterations are effective for acquiring optimized results considering all selected criteria. Here, optimized LM-MLP is the name given to the optimized structure. The value of 500 is set as the primary number of PSO iterations, during which the algorithm converges; the total number of function evaluations includes multiple evaluations per iteration and internal particle update steps, thus being higher than the iteration count. In the present study, the optimal parameters of the PSO algorithm, including the inertia weight (0.7298) and learning coefficients (1.4962), were selected based on reputable and classical studies in particle swarm optimization such as Eberhart and Shi [57]. These values are recommended in references like Clerc and Kennedy [58], which investigated the stability and convergence of PSO, as well as in the research by Poli et al. [59]. Additionally, Engelbrecht [60] presents these settings as standard values. The selection of these parameters aims to achieve an optimal balance between exploration of the search space and exploitation of promising solutions, thereby enhancing the efficiency of the algorithm.

Table 3. PSO's parameters employed in the present research.

Parameter	Value
Inertia weight	0.7298
Swarm size	20
Global learning coefficient	1.4962
Personal learning coefficient	1.4962
Upper limits of variables	6
Lower limits of variables	-6
Stopping iteration	*60000 NFE = 2400 it

*The reported NFE value represents the total number of cost function evaluations throughout the entire optimization process. This value includes not only the evaluations performed during the main iterations but also additional evaluations arising from particle update steps, internal algorithm assessments, and other related procedures. Therefore, this value is considered a flexible overall upper limit that ensures the complete execution and convergence of the algorithm. Accordingly, the observed discrepancy in NFE calculation does not affect the accuracy or quality of the optimization results.

4.3.2. Comparison of predictability of optimized LM-MLP and conventional LM-MLP

The probability distribution for 1000 Monte Carlo simulations of *R* is given in Fig. 11, and Table 4 provides a thorough statistical analysis of the Monte Carlo simulation results. In this study, the performance of the PSO-optimized LM-MLP model significantly improved compared to the conventional model (Table 4). The statistical indicators show that the average RMSE decreased by 5.6%, 21.4%, and 6.1% in the training, testing, and validation datasets, respectively, reflecting higher accuracy of the optimized model. Additionally, the R between predictions and actual data increased by 0.4%, 3.75%, and 1.75% in the respective datasets, indicating a notable improvement in model agreement with the data. These results emphasize that using PSO as a complementary method for optimizing weights and biases can play an effective role in enhancing the accuracy and reliability of neural network models. Furthermore, the reduction in standard deviation values of RMSE and *R* indicates greater stability of the optimized model's performance. Overall, PSO optimization has markedly enhanced the conventional model's predictive performance and increased its generalizability in forecasting the bond strength of corroded reinforcing bars.

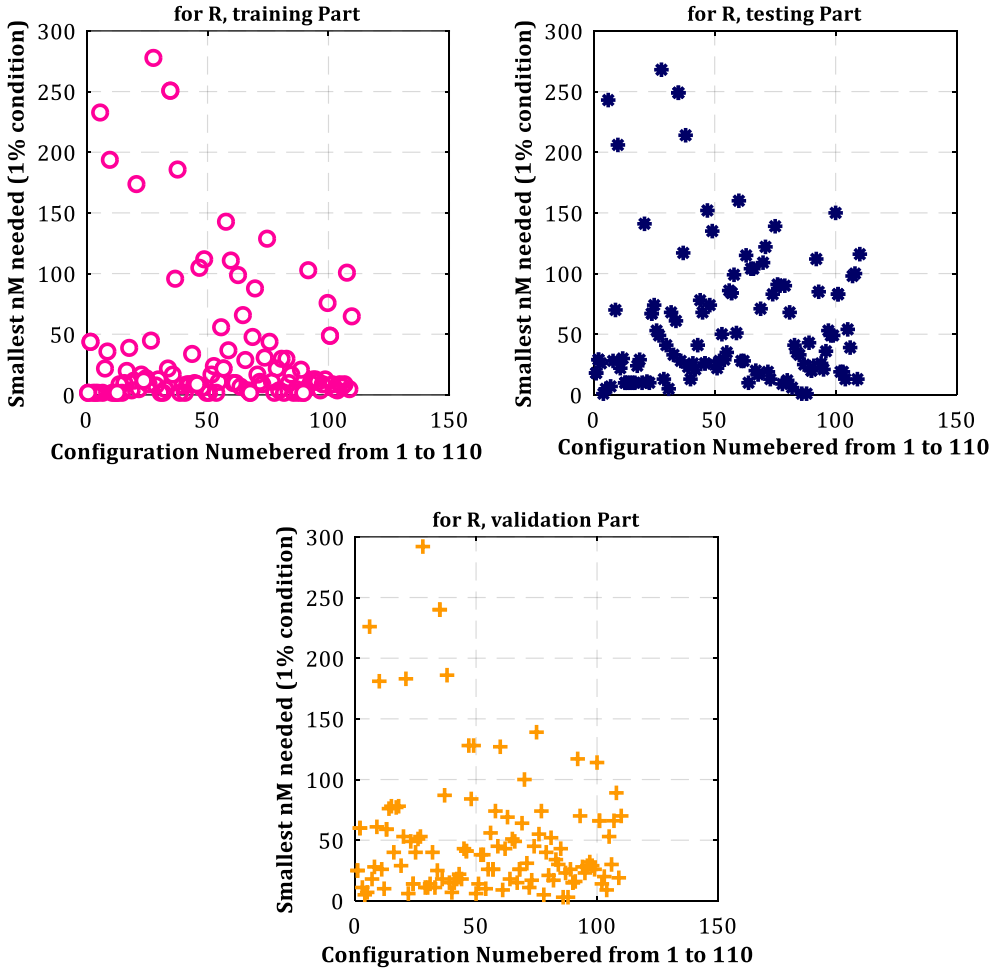


Fig. 5: Least n_M needed for results stabilization within the 1% range of mean values, *R* for different parts of data used in this study.

4.3.3. *Detail performance of optimized LM-MLP in prediction of bond strength*

Here, the typical results of optimized LM-MLP-[4-3-3-1] are shown to prove the capability of prediction by the algorithm. The results of prediction for the training, testing, validation, and whole datasets are shown in Fig. 12 using a regression plot. Regression quality is excellent according to the quality evaluation indicators used in this study; for example, R = 0.953 for the training part, R = 0.968 for the testing part, R = 0.977 for the validation part, and R = 0.959 for the whole data. Additionally, it can be seen that the regression lines for training, testing, validation parts, and whole data had slopes of 0.92, 0.88, 0.97, and 0.92, respectively, between the predicted and actual data. The slopes are near $\pi/4$ (rad), exhibiting an excellent capability of prediction by the optimized LM-MLP-[4-3-3-1]. Generally, this model demonstrated a satisfying performance in forecasting the bond strength of corroded rebar in concrete. All bond strength ranges of corroded rebar in concrete could be covered by the optimized LM-MLP-[4-3-3-1] model. Furthermore, an empirical polynomial relationship is extracted for the datasets on the basis of the classical regression technique and employed to compare with the acquired results from optimized LM-MLP-[4-3-3-1]. In this study, after training the neural network using the LM algorithm, which typically converges to local optima, the PSO algorithm was employed as a metaheuristic method to perform a broader search over the weights and biases space. The purpose of using PSO was to escape local optima and achieve more global optima, leading to improved performance metrics of the model. It is noteworthy that the initial population of PSO was randomly initialized rather than using the weights obtained from the best LM run, ensuring independence and diversity in the search process. This two-stage training approach enhanced the accuracy and generalizability of the model's predictions.

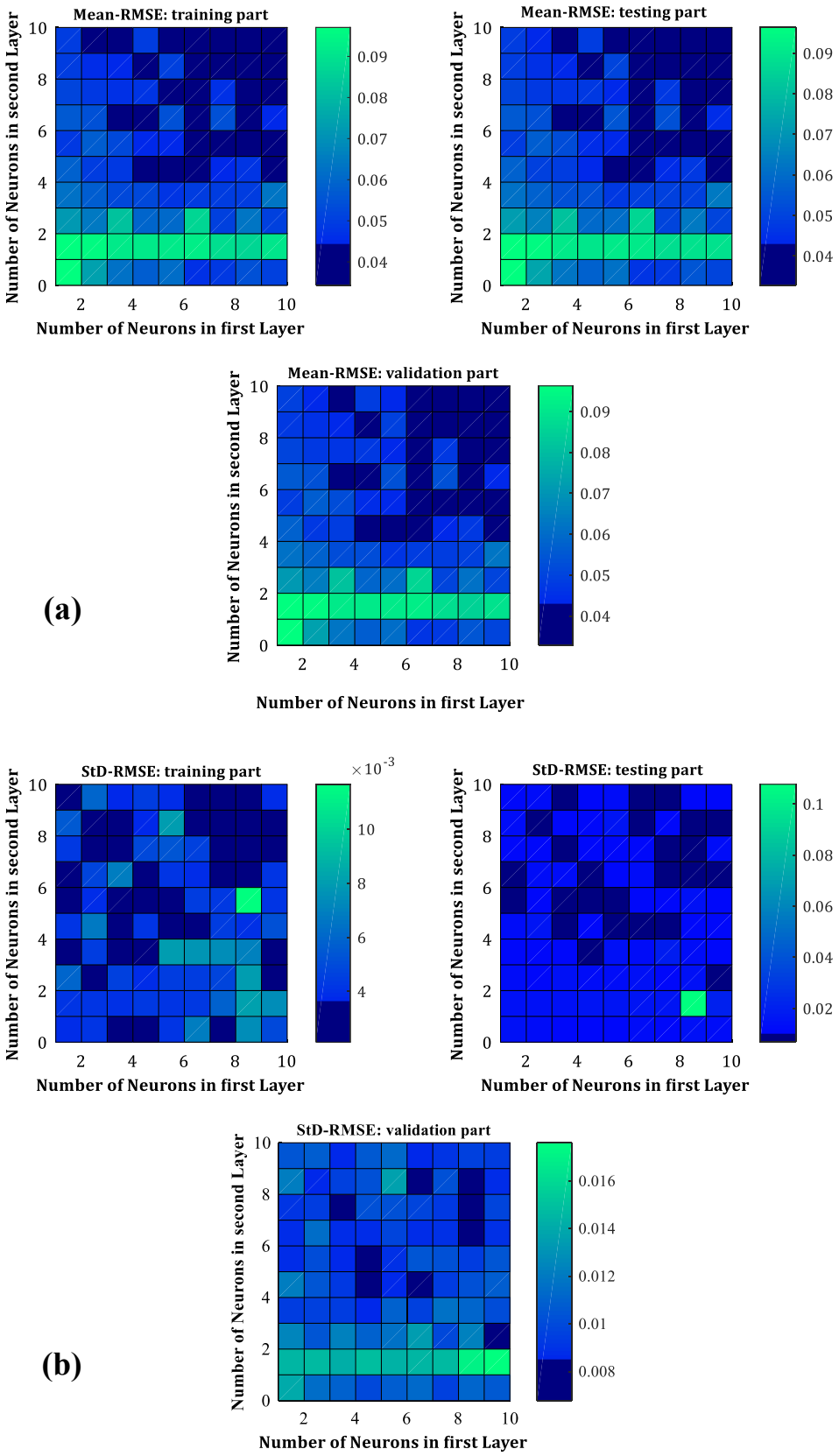


Fig. 6: Parametric investigations considering RMSE criterion for different parts: (a) Mean; (b) StD.

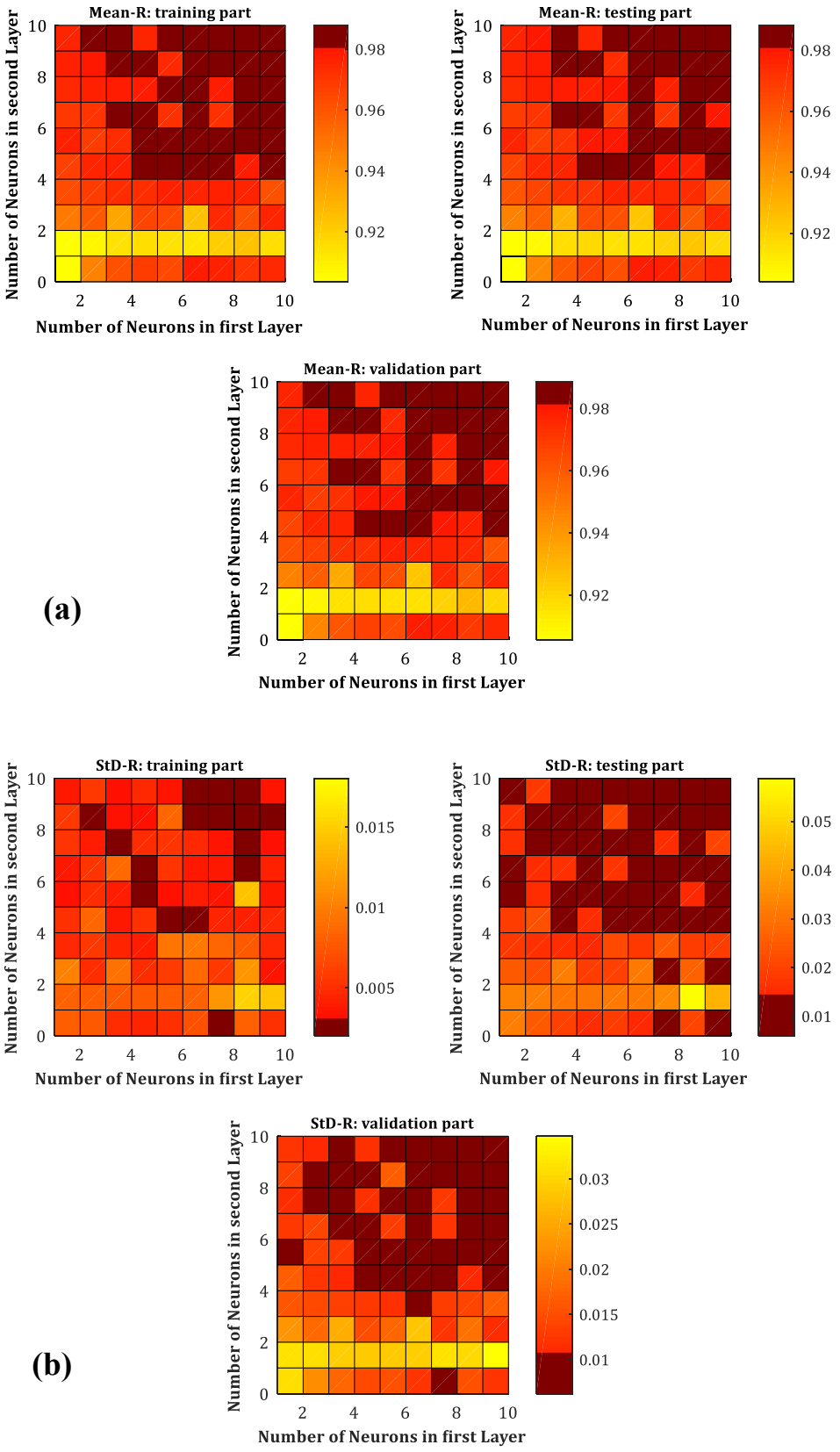


Fig 7: Parametric investigations considering *R* criterion for different parts: (a) Mean; (b) StD.

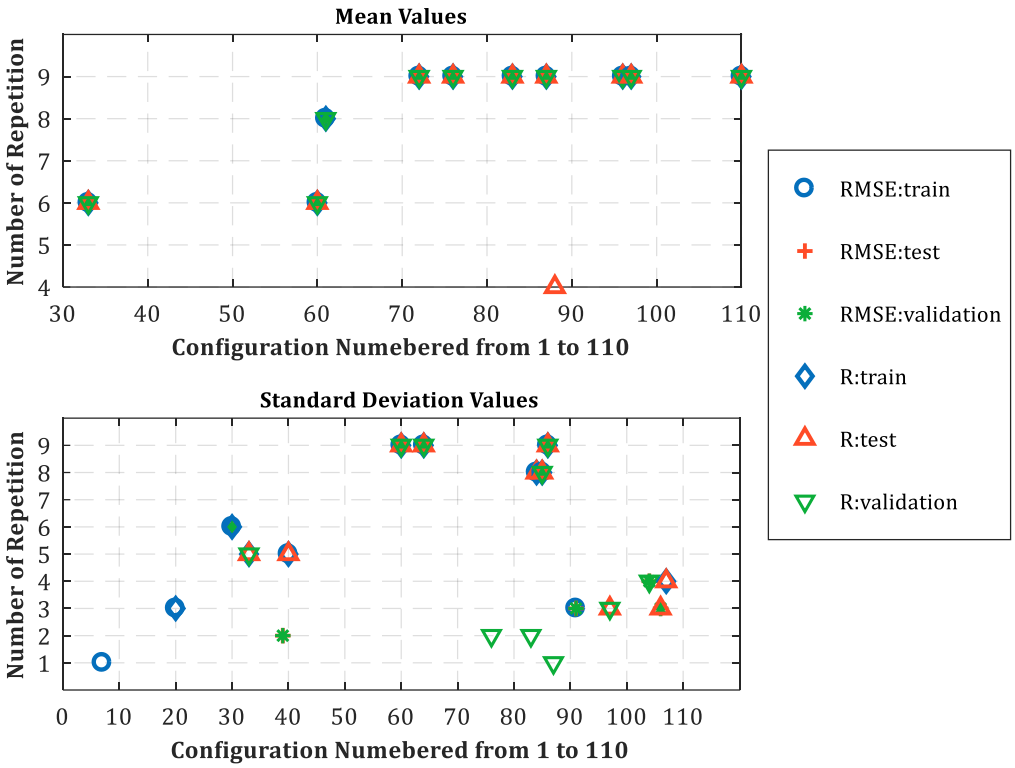


Fig. 8: Top ten configurations with the lowest mean and standard deviation values.

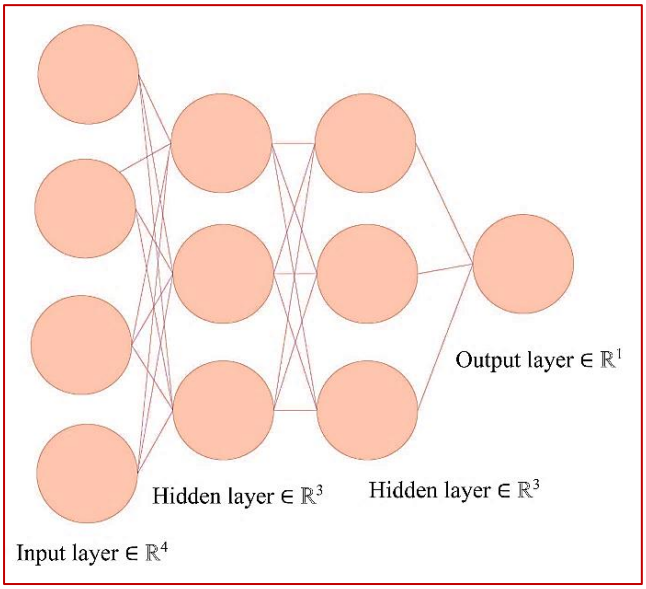


Fig. 9: Optimized LM-MLP in this study.

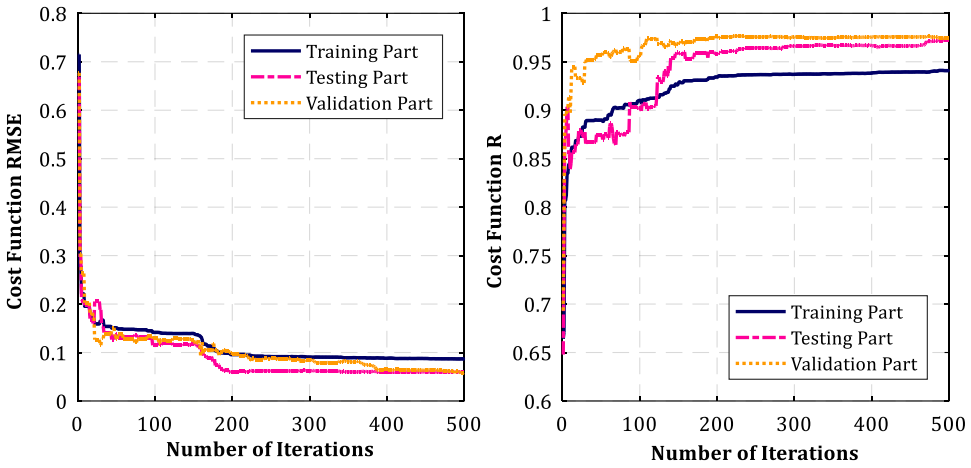


Fig. 10: Evolution of *R* and RMSE during the optimization procedure.

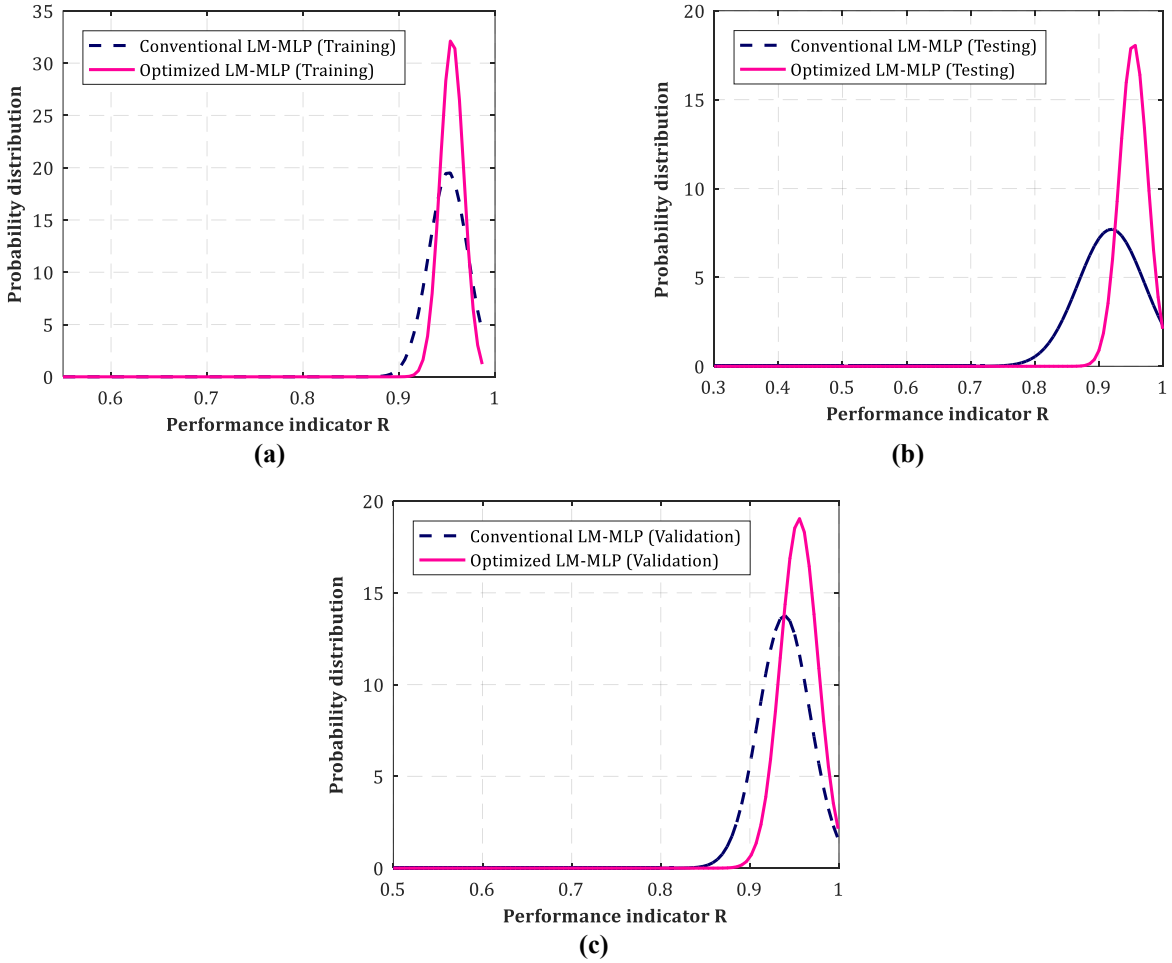


Fig. 11: Statistical performance comparison of 1000 *R* simulations for: (a) training dataset; (b) testing dataset; (c) validation dataset.

Since normalized input and output variables were employed in the optimized LM-MLP-[4-3-3-1] model, for an accurate comparison, the following equation developed based on the classical regression approach is also obtained considering the normalized variables (with the subscript n):

$$u_n = 0.4264103 + 0.70166169t_{cc,n} - 0.38255386m_{l,n} - 0.96424809w_{c,n} - 0.02514181C_{Chl,n} - 0.31683072{t_{cc,n}}^2 + 0.8228890{w_{c,n}}^2 \quad ; \quad R = 0.922 \tag{5}$$

Fig. 13 shows the regression diagram comparing the forecasted and experimental values of the bond strength of corroded rebar in concrete employing such a relationship. It is seen that a good correlation value (R = 0.922) is acquired but not as good as the value (R = 0.959) obtained from the optimized LM-MLP-[4-3-3-1].

The example demonstrates how to take real input values, normalize them, calculate the normalized output via the regression equation, and then denormalize it to obtain bond strength in physical units (Appendix A). This guidance facilitates the practical use of the normalized model for accurate bond strength prediction.

The conducted sensitivity analysis indicates that the model's input parameters significantly affect the bond strength of corroded reinforcing bars in concrete, which aligns well with previous observations and findings in structural engineering and corrosion research (Table 5). Specifically, increased corrosion levels and chloride penetration directly lead to a reduction in bond adhesion and strength, consistent with the detrimental effects of corrosion on structural durability. Conversely, increased concrete cover thickness provides enhanced protective performance, improving bond strength. Additionally, increased crack width, resulting from corrosion propagation and mechanical stresses, signals structural deterioration that reduces the load-bearing capacity of the bond. These findings emphasize the critical importance of controlling environmental factors and concrete materials to maintain the durability and safety of reinforced concrete structures. This information serves as a valuable guide for civil engineers in optimizing the design and maintenance of structures under corrosive conditions.

Table 4. Statistical performance of RMSE and R for conventional LM-MLP and optimized LM-MLP, including percentage improvement.

Criterion	Model	Training Mean	Training StD	Testing Mean	Testing StD	Validation Mean	Validation StD
RMSE	Conventional LM-MLP	0.07331	0.00020	0.09125	0.00149	0.07679	0.00025
	Optimized LM-MLP	0.06920	0.00012	0.07170	0.00026	0.07209	0.00025
	Improvement (%)	5.60%	—	21.42%	—	6.12%	—
R	Conventional LM-MLP	0.95132	0.00041	0.91929	0.00268	0.93883	0.00084
	Optimized LM-MLP	0.95508	0.00015	0.95379	0.00048	0.95521	0.00044
	Improvement (%)	0.40%	—	3.75%	—	1.75%	—

Caption:
Percentage improvement for RMSE is calculated as
$\frac{RMSE_{Conventional} - RMSE_{\text{Optimized}}}{RMSE_{Conventional}} \times 100$
indicating the percent decrease in error for the optimized model.
For the correlation coefficient (R), percentage improvement is computed as
$\frac{R_{Optimized} - R_{Conventional}}{R_{Conventional}} \times 100$
showing the percent increase in correlation for the optimized model.

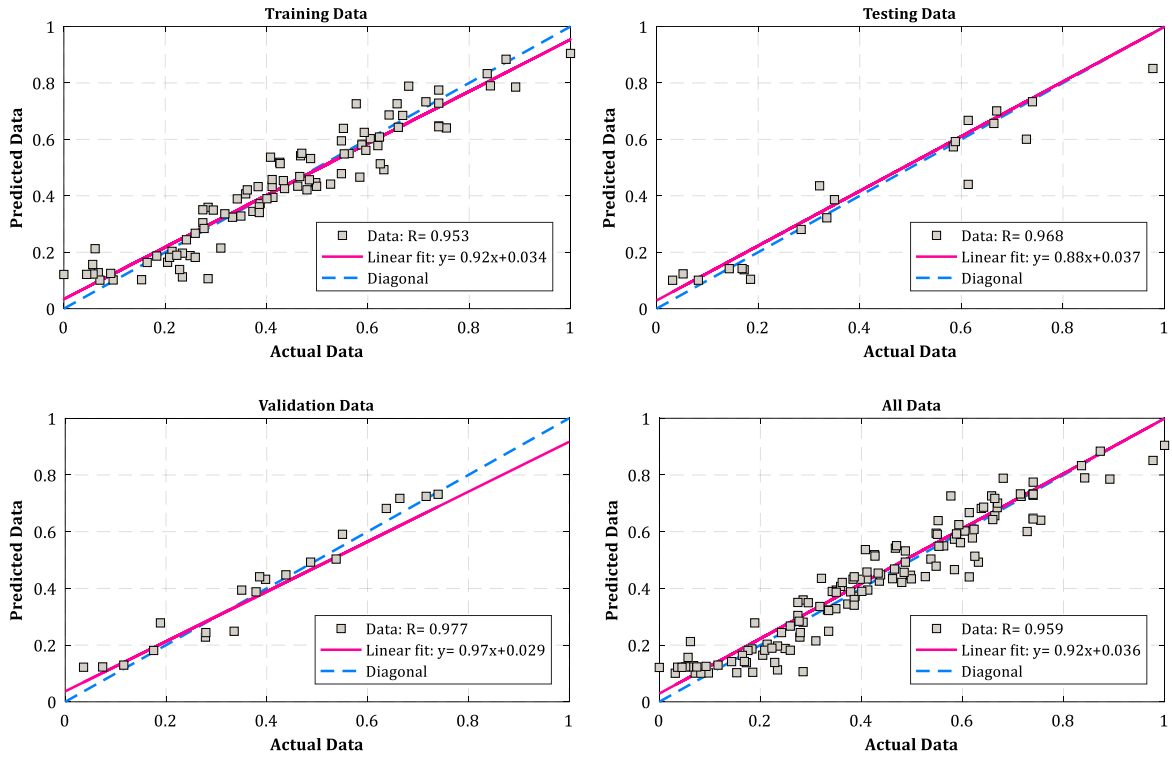


Fig. 12: Regression plots for training, testing, validation and whole data by Optimized LM-MLP.

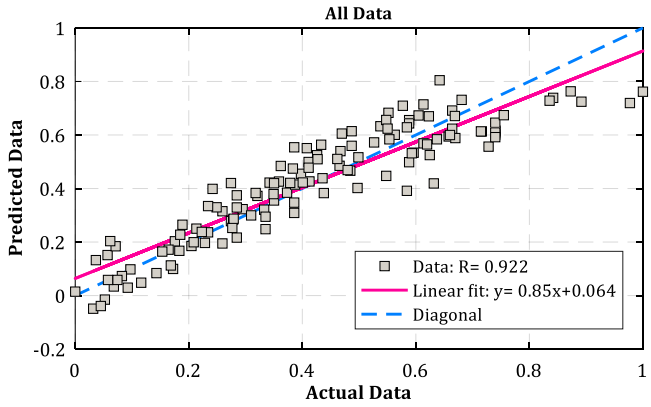


Fig. 13: Regression plots for predicted data employing Eq. (5) and actual data of bond strength.

5. Conclusions

Although some research studies have been carried out to forecast the bond strength of corroded rebar in concrete with different neural networks or regression methods, the accuracy and robustness of these methods still require further comprehensive examination. This paper presents a thorough and coherent procedure to accurately forecast the bond strength of corroded rebar in concrete employing a Levenberg-Marquardt (LM)-based Multi-Layer Perceptron (MLP) neural network optimized with PSO (optimized LM-MLP). The outcomes of this study can be summarized as follows:

- Based on investigating the LM-MLP structure, it can be deduced that: (i) utilizing two hidden layers can give better forecast accuracy than utilizing a single hidden layer, and (ii) increasing the number of neurons in both hidden layers up to the maximum number of neurons in this study assures the increase of LM-MLP accuracy.
- Utilizing the statistical analysis and the Monte Carlo simulations for decision-making led to finding the optimal architecture of LM-MLP- [4-3-3-1] as the best

predictor even considering the random generation process of the dataset.

- The optimized LM-MLP structure using the PSO algorithm was proven to create comparative results and an acceptable correlation coefficient of R = 0.959.
- The comparison of the prediction capability of optimized LM-MLP and conventional LM-MLP showed that optimized LM-MLP was more accurate than conventional LM-MLP.
- Generally, the optimized LM-MLP-[4-3-3-1] model demonstrated a satisfying performance in forecasting the bond strength of corroded rebar in concrete. This model is capable of capturing the full range of bond strength values of corroded rebar in concrete.
- Also, an empirical polynomial relationship was extracted for the dataset on the basis of the classical regression technique and utilized to compare with the results acquired from optimized LM-MLP-[4-3-3-1]. It was seen that a good correlation value (R = 0.922) is achieved but not as good as the value (R = 0.959) acquired from the optimized LM-MLP-[4-3-3-1].
- The results indicate the superior performance of the PSO-optimized LM-MLP model, demonstrating an increase of up to 3.75% in the correlation coefficient and a reduction of up to 21.42% in RMSE compared to the conventional LM-MLP model. These improvements emphasize the effectiveness of the proposed method in more accurately predicting the bond strength of corroded reinforcing bars and enhance its performance over classical approaches.

6. Limitations and Future Work

While the proposed ANN model optimized via Monte Carlo simulation and PSO demonstrates improved predictive capability for the bond strength of corroded rebars, several limitations should be acknowledged. First, the dataset used is relatively small and derived from a single experimental program, which may restrict the generalizability of the model to other conditions and datasets. Second, the model's validation is limited to the available data, lacking external validation on independent experimental datasets. Third, the parameters such as bar diameter, concrete strength, and anchorage length were kept constant, limiting the model's applicability across diverse structural scenarios. Future research should consider integrating larger and more diverse datasets and evaluating alternative machine learning algorithms to enhance robustness and wide applicability. Additionally, applying transfer learning or domain adaptation techniques may further improve model generalization.

Table 5. Sensitivity of input parameters on bond strength and experimental consistency.

Input Parameter	Parameter Change	Effect on Bond Strength	Explanation and Consistency with Experimental Knowledge
Mass Loss	Increase	Significant decrease	Increased corrosion reduces bond adhesion between rebar and concrete, leading to reduced strength, consistent with studies such as Amleh [20] and Almusallam et al. [18].
Cover Thickness	Increase	Increase	Thicker concrete cover provides better protection to the rebar and enhances bond strength, in agreement with durability research findings.
Crack Width	Increase	Decrease	Increased crack width due to corrosion propagation and mechanical damage reduces bond performance, aligning with validated experimental data.
Chloride Content	Increase	Decrease	Chloride penetration weakens the protective layer of rebar and accelerates corrosion, resulting in reduced bond strength; this effect is confirmed by multiple studies.

Appendix A : Step-by-step example of normalization, regression calculation, and denormalization for bond strength prediction

Step	Description	Formula / Calculation	Numerical Value
1	Given real input values	CT=40 mm, ML=5%, CW=0.2 mm, Cl=1.5%	-
2	Normalize input values	$Z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$	CT:0.2, ML:0.4, CW:0.333, Cl:0.378
3	Calculate normalized output u	u=0.4246103+0.70166169×0.2−0.38255386×0.4−0.96424809×0.333−0.02514181×0.378 −0.31683072×0.3782+0.8228890×0.333×0.378	u=0.1404
4	Denormalize output	y=u×(4.34−0.57)+0.57=1.10	y=1.10

Acknowledgements

This work has been financially supported by the Vice-Chancellor for Research of the University of Torbat-e Jam. The authors express their sincere gratitude for this support, which has been instrumental in facilitating the research presented in this paper.

References

[1] Turan Aİ, Ayaz Y, Yalciner H, Kumbasaroglu A. An experimental evaluation on structural performance level of corroded reinforced concrete frames. Engineering Structures. 2025;325:119479.
[2] Song H-W, Saraswathy V. Corrosion monitoring of reinforced concrete structures-A. Int J Electrochem Sci. 2007;2:1-28.
[3] Lin H, Zhao Y, Feng P, Ye H, Ozbolt J, Jiang C et al. State-of-the-art review on the bond properties of corroded reinforcing steel bar. Construction and Building Materials. 2019;213:216-33.
[4] Hughes B, Videla C. Design criteria for early-age bond strength in reinforced concrete. Materials and Structures. 1992;25:445-63.
[5] Cairns J, Abdullah R. Evaluation of bond pullout tests and their relevance to structural performance. The Structural Engineer. 1995;73:179-85.
[6] Dahou Z, Sbartaï ZM, Castel A, Ghomari F. Artificial neural network model for steel–concrete bond prediction. Engineering Structures. 2009;31:1724-33.
[7] Cairns J, Jones K. Influence of rib geometry on strength of lapped joints: an experimental and analytical study. Magazine of Concrete Research. 1995;47:253-62.
[8] Andrade MSA, Ribeiro JCL, de Oliveira DS, Pedroti LG, Santos CFR. Experimental evaluation of concrete-reinforcement bond: Bond failure mechanisms after exposure to elevated temperatures. Engineering Structures. 2024;311:118148.
[9] Gambarova PG, Rosati GP. Bond and splitting in bar pull-out: behavioural laws and concrete cover role. Magazine of Concrete Research. 1997;49:99-110.
[10] Luccioni BM, López DE, Danesi RF. Bond-slip in reinforced concrete elements. Journal of structural engineering. 2005;131:1690-8.
[11] Alhawat M, Ashour A. Bond strength between corroded steel reinforcement and recycled aggregate concrete. Structures: Elsevier; 2019. p. 369-85.
[12] Auyeung Y, Balaguru P, Chung L. Bond behavior of corroded reinforcement bars. Materials Journal. 2000;97:214-20.
[13] Horrigmoe G, Saether I, Antonsen R, Arntsen B. Laboratory investigations of steel bar corrosion in concrete: Sustainable Bridges Background document SB3. 10. 2007.
[14] Yalciner H, Eren O, Sensoy S. An experimental study on the bond strength between reinforcement bars and concrete as a function of concrete cover, strength and corrosion level. Cement and Concrete Research. 2012;42:643-55.
[15] Güneyisi EM, Mermerdaş K, Gültekin A. Evaluation and modeling of ultimate bond strength of corroded reinforcement in reinforced concrete elements. Materials and Structures. 2016;49:3195-215.
[16] Hoang N-D, Tran X-L, Nguyen H. Predicting ultimate bond strength of corroded reinforcement and surrounding concrete using a metaheuristic optimized least squares support vector regression model. Neural Computing and Applications. 2020;32:7289-309.
[17] Saifullah M, Clark L. Effect of corrosion rate on the bond strength of corroded reinforcement. Corrosion and corrosion protection of steel in concrete. 1994;1:591-602.
[18] Almusallam AA, Al-Gahtani AS, Aziz AR. Effect of reinforcement corrosion on bond strength. Construction and building materials. 1996;10:123-9.
[19] Fang C, Lundgren K, Chen L, Zhu C. Corrosion influence on bond in reinforced concrete. Cement and concrete research. 2004;34:2159-67.
[20] Amleh L. Bond deterioration of reinforcing steel in concrete due to corrosion [PhD Thesis]. Montreal: McGill University; 2000.
[21] Kranc S, Sagüés AA. Detailed modeling of corrosion macrocells on steel reinforcing in concrete. Corrosion Science. 2001;43:1355-72.
[22] Steffens A, Dinkler D, Ahrens H. Modeling carbonation for corrosion risk prediction of concrete structures. Cement and Concrete Research. 2002;32:935-41.
[23] Parthiban T, Ravi R, Parthiban G, Srinivasan S, Ramakrishnan K, Raghavan M. Neural network analysis for corrosion of steel in concrete. Corrosion Science. 2005;47:1625-42.
[24] Karray F, Karray FO, De Silva CW. Soft computing and intelligent systems design: theory, tools, and applications: Pearson Education; 2004.
[25] Ai D, Mo F, Han Y, Wen J. Automated identification of compressive stress and damage in concrete specimen using convolutional neural network learned electromechanical admittance. Engineering Structures. 2022;259:114176.
[26] Elshafey AA, Dawood N, Marzouk H, Haddara M. Crack width in concrete using artificial neural networks. Engineering Structures. 2013;52:676-86.
[27] Köroğlu MA, Ceylan M, Arslan MH, Ilki A. Estimation of flexural capacity of quadrilateral FRP-confined RC columns using combined artificial neural network. Engineering Structures. 2012;42:23-32.
[28] Tanyildizi H. Predicting bond strength of corroded reinforcement by deep learning. Computers and Concrete. 2022;29:145-59.
[29] Shirkhani A, Davarnia D, Azar BF. Prediction of bond strength between concrete and rebar under corrosion using ANN. Computers and Concrete, An International Journal. 2019;23:273-9.
[30] Ikumi T, Galeote E, Pujadas P, de la Fuente A, López-Carreño R. Neural network-aided prediction of post-cracking tensile strength of fibre-reinforced concrete. Computers & Structures. 2021;256:106640.
[31] Ghorbani B, Arulrajah A, Narsilio G, Horpibulsuk S, Bo MW. Hybrid Formulation of Resilient Modulus for Cohesive Subgrade Soils Utilizing CPT Test Parameters. Journal of Materials in Civil Engineering. 2020;32:06020011.
[32] Pan L, Novák L, Lehký D, Novák D, Cao M. Neural network ensemble-based sensitivity analysis in structural engineering: Comparison of selected methods and the influence of statistical correlation. Computers & Structures. 2021;242:106376.
[33] Mate C, Jimeńez L. Forecasting exchange rates with the iMLP: New empirical insight on one multi-layer perceptron for interval time series (ITS). Engineering Applications of Artificial Intelligence. 2021;104:104358.
[34] Ly H-B, Nguyen MH, Pham BT. Metaheuristic optimization of Levenberg–Marquardt-based artificial neural network using particle swarm optimization for prediction of foamed concrete compressive strength. Neural Computing and Applications. 2021;33:17331-51.
[35] Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of clinical epidemiology. 1996;49:1225-31.
[36] Deo RC, Ghorbani MA, Samadianfard S, Maraseni T, Bilgili M, Biazar M. Multi-layer perceptron hybrid model integrated with the firefly optimizer algorithm for windspeed prediction of target site using a limited set of neighboring reference station data. Renewable energy. 2018;116:309-23.

[37] Yeh I-C. Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete research. 1998;28:1797-808.
[38] Topçu İB, Sarıdemir M. Prediction of rubberized concrete properties using artificial neural network and fuzzy logic. Construction and Building Materials. 2008;22:532-40.
[39] Trtnik G, Kavčič F, Turk G. Prediction of concrete strength using ultrasonic pulse velocity and artificial neural networks. Ultrasonics. 2009;49:53-60.
[40] Dias W, Pooliyadda S. Neural networks for predicting properties of concretes with admixtures. Construction and Building Materials. 2001;15:371-9.
[41] Pradhan B, Lee S. Landslide susceptibility assessment and factor effect analysis: backpropagation artificial neural networks and their comparison with frequency ratio and bivariate logistic regression modelling. Environmental Modelling & Software. 2010;25:747-59.
[42] Delnavaz M, Ayati B, Ganjidoust H. Prediction of moving bed biofilm reactor (MBBR) performance for the treatment of aniline using artificial neural networks (ANN). Journal of hazardous materials. 2010;179:769-75.
[43] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proceedings of the sixth international symposium on micro machine and human science: Ieee Piscataway, NJ, USA; 1995. p. 39-43.
[44] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering applications of artificial intelligence. 2007;20:89-99.
[45] Yildiz AR. A new hybrid particle swarm optimization approach for structural design optimization in the automotive industry. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2012;226:1340-51.
[46] Alam MN. Particle swarm optimization: Algorithm and its codes in matlab. ResearchGate. 2016;8:1-10.
[47] Pathak NN, Mahanti G, Singh SK, Mishra JK, Chakraborty A. Synthesis of thinned planar circular array antennas using modified particle swarm optimization. Progress In Electromagnetics Research Letters. 2009;12:87-97.
[48] Qi C, Ly H-B, Chen Q, Le T-T, Le VM, Pham BT. Flocculation-dewatering prediction of fine mineral tailings using a hybrid machine learning approach. Chemosphere. 2020;244:125450.
[49] Yong W, Zhou J, Jahed Armaghani D, Tahir M, Tarinejad R, Pham BT et al. A new hybrid simulated annealing-based genetic programming technique to predict the ultimate bearing capacity of piles. Engineering with Computers. 2021;37:2111-27.
[50] Pham BT, Le LM, Le T-T, Bui K-TT, Le VM, Ly H-B et al. Development of advanced artificial intelligence models for daily rainfall prediction. Atmospheric Research. 2020;237:104845.
[51] Ly H-B, Le T-T, Vu H-LT, Tran VQ, Le LM, Pham BT. Computational hybrid machine learning based prediction of shear capacity for steel fiber reinforced concrete beams. Sustainability. 2020;12:2709.
[52] Topçu İB, Boğa AR, Hocaoğlu FO. Modeling corrosion currents of reinforced concrete using ANN. Automation in Construction. 2009;18:145-52.
[53] Gao P, Li Y, Sun J, Huang GH. A Monte-Carlo-based interval De Novo programming method for optimal system design under uncertainty. Engineering Applications of Artificial Intelligence. 2018;72:30-42.
[54] Green II RC, Wang L, Alam M, Singh C. Intelligent state space pruning for Monte Carlo simulation with applications in composite power system reliability. Engineering applications of artificial intelligence. 2013;26:1707-24.
[55] Martín-Fernández L, Ruiz DP, Torija AJ, Míguez J. A Bayesian method for model selection in environmental noise prediction. Journal of Environmental Informatics. 2016;27:31-42.
[56] Yu L, Li Y, Huang GH, Shan B. A hybrid fuzzy-stochastic technique for planning peak electricity management under multiple uncertainties. Engineering Applications of Artificial Intelligence. 2017;62:252-64.
[57] Eberhart R, Shi Y. Particle swarm optimization: developments, applications and resources. Proceedings of the 2001 congress on evolutionary computation (IEEE Cat No 01TH8546): IEEE; 2001. p. 81-6.
[58] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE transactions on Evolutionary Computation. 2002;6:58-73.
[59] Poli R, Kennedy J, Blackwell T. Particle swarm optimization: An overview. Swarm intelligence. 2007;1:33-57.
[60] Engelbrecht AP. Fundamentals of Computational Swarm Intelligence. Hoboken, NJ: Wiley. 2005.