1. Introduction

Artificial neural networks (ANNs) continue to gain interest in many engineering fields such as civil engineering, developments of construction materials, agriculture applications, etc. Iqbal et al. (2020) investigated the ability of re-useage of waste foundry sand (WFS) for development of green concrete. They successfully predicted the compressive strength, elastic modulus and tensile strength of concrete made from WFS based on gene expression programming (GEP). ANNs and GEP were also implemented by Martí et al. (2013) in estimating dissolved oxygen (DOₘ) in micro-irrigation sand filters fed with effluents; and by Azimi-Pour and Eskandari-Naddaf (2018) in predicting strength of cement mortar considering effects of nano and micro silica. Moreover, Gandomi et al. (2011) utilized GEP to predict the flow number of dense asphalt-aggregate mixtures. Srinivasulu and Jain (2006) compared training methods, including the back-propagation algorithm (BPA), the real-coded genetic algorithm (RGA), and a self-organizing map (SOM), which were developed to train the multilayer perceptrons (MLPs) of ANNs for modeling the rainfall–runoff process. They found that the RGA-trained ANN model substantially
outperforms the BPA-trained ANN model, with the ability to overcome some limitations of the BPA-trained ANN rainfall–runoff model. Moreover, they showed that the RGA-trained ANN rainfall–runoff model achieves a better generalization of the complex, dynamic, nonlinear, and fragmented rainfall–runoff process compared to the other approaches.

The perceptron training rule works well when the samples to be trained are linearly separable while updating weights based on the backpropagation errors. In contrast, the Adaline rule works well even when the samples to be trained are not linearly separable while converging toward the best-fit approximation of the target output. While the training methods work well for single-layer neural networks or perceptrons, the present paper proposes efficient training schemes that will work for ANNs with both shallow and deep layers.

Numerous structural issues, including those related to the analysis and design, were addressed by Mukherjee and Deshpande (1995), Azim et al. (2020a, 2020b, 2021), Abdalla and Stavroulakis (1995), Chen et al. (1995), Messner, Sanvido, and Kumara (1994), Kang and Yoon (1994), Isa and Fletcher (1993). However, according to Gupta and Sharma (2011), the weak performance and numerical instability encountered in training big structural models prevents the use of neural networks. Hong (2019) investigated the influence of using several layers and neurons on the training accuracy of designing doubly reinforced concrete beams. The present study attempts to overcome the predicament related to ANNs that are stimulated to learn trends of large structural design datasets, by mapping the input variables to output variables for engineering applications. CTS and CRS were developed to improve training accuracies as well as enhance the ability to accurately design structural problems based on CRS- and CTS-trained ANN networks.

Generally, multiple input variables are directly mapped to output variables while training ANNs with shallow or deep layers. This training, which involves mapping all input parameters to all output parameters, is easy and simple. However, such a training yielded insufficient accuracy when designing complex structural systems such as doubly reinforced beams, where seven input parameters ($M_{in}$, $L$, $f_y$, $f_o$, $d$, $ConUP$, and $ReUP$) were selected as input feature indexes to predict four outputs [$b$, $\rho_o$, $\rho_o$, and cost of beam materials (CBM)].

In this study, parallel training method (PTM) maps input vector to each output parameter based on individual neural network, resulting in implementing a numbers of ANN equal to a numbers of output parameters. The chained training scheme (CTS) and CTS with a revised sequence (CRS) are also used to train ANNs with deep layers for designing doubly reinforced concrete beams, to decrease the mean square error (MSE) and increase the $R$ value. The CRS and CTS methods are based on the feature selection scores determined by the neighborhood component analysis (NCA) when training ANNs on large datasets. In addition, the training sequence of the input feature indexes for CRS is determined as per NCA’s recommendation; however, the input feature indexes are extended to capture better training accuracies. CTS and CRS design concrete beams much more efficiently than conventional training on entire data (TED) method. A design can be commenced selecting using any variables which can be treated as artificial neural genes. CRS and CTS methods were based on feature selection scores determined by Neighborhood Component Analysis (NCA) when training ANNs. Training sequence of input feature indexes for CRS was determined recommended by NCA, but input feature indexes were extended to capture better training accuracies. The proposed methods can be extended to broad areas of structural engineering designs for practical applications.

![Figure 1. Comparison between DL and conventional ML algorithms (Shrestha, Krishna, and Von Krogh 2021).](image)

2. Common machine learning (ML) methods vs. ANNs with deep layers

Machine learning (ML) provides mathematical optimization based on the regression predicted by multiple ML models using computational statistics. ML does not require establishing networking parameters, such as neurons, layers, and epochs, which need to be carefully selected when training deep networks. A mathematical model of structural data was built by ML algorithms to predict the structural behaviors and design based on 19 train regression models shown in Regression Learner App, MATLAB (2020b), including linear regression models, regression trees, Gaussian process regression models, support vector machines, and ensembles of regression trees. Then, the best ML model was selected for each output parameter, avoiding overfitting by applying cross-validation based on a validation scheme when ML models were trained. ML is considered better than
neural networks with deep layers when the number of datasets is small, requiring less computation than DL. Conventional ML algorithms such as Linear Regressions, Regression Trees, Support Vector Machine, Gaussian Process Regression, etc. approximate functions of data using various strategies namely least-squares fit techniques (Linear Regressions), statistic procedures (SVM), or kernel-based probabilistic solutions (Gaussian Process Regression), etc. These methods, although, show

**Figure 2.** Topology of artificial neural network (ANN).

**Figure 3.** Procedure to generate big datasets based on Autobeam.
adequate accuracies in small numbers of data, their fitting capabilities are limited when numbers of data increase as shown in Figure 1. Deep neural networks, on the other hand, theoretically are capable of simulating any continuous functions when sufficient numbers of neurons and layers are given, outperforming conventional ML algorithms in dealing with large datasets. Shrestha, Krishna, and Von Krogh (2021) stated that DL has a better performance to deal with massive data than conventional ML methods as shown in Figure 1. Liu et al. (2021) also conducted a comparison of performance of three different ANN models considering different database conditions.

3. ANN model and big data generation

There are seven inputs \( (M_n, L, f_p, f_p', d, \text{ConUP}, \text{and ReUP}) \) and four outputs \( (b, \rho_b, \rho_o, \text{and CBM}) \) for reverse design of doubly RC beam considered in this study. Figure 2 illustrates the topology of deep neural network to predict beam width \( (b) \), both tensile \( (\rho_o) \) and compressive rebar ratio \( (\rho_c) \), and cost of beam materials \( (\text{CBM}) \) in an output side based on input side including nominal moment capacity \( (M_n) \), beam length \( (L) \), beam depth \( (d) \), material properties (yield rebar strength, \( f_y \) and compressive concrete strength, \( f_c \)), and concrete \( (\text{ConUP}) \) and rebar \( (\text{ReUP}) \) unit price.

An analytical software entitled as Autobeam developed by Nguyen and Hong (2019) was used to generate 20,000 datasets. The procedure to generate big datasets based on Autobeam software was illustrated as Figure 3. The minimum limit of tensile rebar ratio is controlled as a function of compressive concrete strength \( (f_c') \) and yield rebar strength \( (f_y) \) following requirements of Section 9.6.1.2, ACI (2019) 318-19. Then, 20,000 datasets were divided into three datasets which are training, validation, and testing data with the proportions of 70%, 15%, and 15%, respectively. The statistical parameters (maximum, median, mean, minimum, standard deviation, variance) of 20,000 datasets are shown in Table 1.

4. Feature selection scores

Several methods have been introduced in MATLAB for feature selection, such as F-test and NCA. The F-test weighted by the probability values \( (P\text{-values}) \) examines two hypotheses \( (Table \text{2(a)}) \): null and alternative hypotheses. Under varying input, the mean values of the output remain the same for the null hypothesis, whereas those of the output are not the same for the alternative hypothesis. NCA \( (Table \text{2(b)}) \) predicts the unseen data based on the closest points that it has learned. A simple distance function between A \( (x_1, y_1, z_1) \) and B \( (x_2, y_2, z_2) \) can be derived as

\[
dw = |x_1 - x_2| + |y_1 - y_2|
\]

where \( x \) and \( y \) are the inputs, whereas \( z \) is the output feature, which is omitted since it is output.

However, distance treats all inputs equally. Therefore, weight factors are added to emphasize the key features:

\[
dw = w_1^2|x_1 - x_2| + w_2^2|y_1 - y_2|
\]

The probability that each data point has similar prediction output (randomized regression model) is maximized based on its closest points (shortest distances) by adjusting the weight factors by using the NCA feature selection function.

The probability that each data point has similar prediction output (randomized regression model) is maximized based on its closest points (shortest distances) by adjusting the weight factors by using the NCA feature selection function. The RRelieff algorithm is used to score features. This method
Table 2. Feature selection scores a) Feature selection scores based on F-test. b) Feature selection scores based on NCA method.

### Feature selection based on 20,000 datasets using F-test method

| Input | Output | $M_n$ | $L$ | $f_p$ | $f_z$ | $d$ | ConUP | RelUP | $b$ | $p_t$ | $p_c$ | CBM |
|-------|--------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|
| $b$   |        | Inf   | 3.13| 4.30  | 0.28  | 1.21| 0.28  | 4.30  | 0.50| 0.22  | Inf   |     |
| $p_t$ |        | Inf   | 0.46| Inf   | Inf   | 0.37| Inf   | Inf   | Inf | Inf   | Inf   |     |
| $p_c$ |        | Inf   | 0.26| 289.74| 573.17| 1.07| 573.17| 289.74| 0.28| Inf   | Inf   |     |
| $CBM$ |        | Inf   | 2.13| 103.51| Inf   | Inf | Inf   | Inf   | Inf | Inf   | Inf   |     |

### Feature selection based on 20,000 datasets using NCA method

| Input | Output | $M_n$ | $L$ | $f_p$ | $f_z$ | $d$ | ConUP | RelUP | $b$ | $p_t$ | $p_c$ | CBM |
|-------|--------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|
| $b$   |        | 1.93E-56| 2.06E-149| 3.86E-66| 4.80  | 9.49| 2.12E-08| 1.59E-76| 9.02| 3.54  | 16.39 |     |
| $p_t$ |        | 6.06  | 5.14E-322| 2.98   | 3.37  | 8.55| 3.36E-211| 7.7E-230 | 7.43| 4.07  | 12.24 |     |
| $p_c$ |        | 2.89E-53| 1.02E-45  | 5.11   | 4.593.76E-451.36E-073.19E-31| 6.91E-478.14| 9.31E-45|     |
| $CBM$ |        | 5.71  | 2.62E-321| 1.61E-155| 3.76  | 6.04| 5.16E-431| 0.3E-297 | 6.36| 6.221.77|     |
Table 3. Training accuracies based on TED, PTM, CTS, and CRS.

(a) TED

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 3.718                  | 4,218        | 1.03E-02 | 0.876           |
| 2   | 20,000  | 50     | 50      | 50,000         | 3.446                  | 3,946        | 1.04E-02 | 0.877           |
| 3   | 50,000  | 40     | 40      | 50,000         | 45.45                 | 5,045        | 9.84E-03 | 0.880           |
| 4   | 50,000  | 50     | 50      | 50,000         | 7.204                  | 7,704        | 9.89E-03 | 0.880           |
| 5   | 50,000  | 60     | 60      | 50,000         | 8,292                  | 8,792        | 1.06E-02 | 0.870           |

(b) PTM

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 10,419                 | 10,919       | 2.96E-04 | 0.984           |
| 2   | 50,000  | 50     | 50      | 50,000         | 7,365                  | 7,865        | 2.44E-04 | 0.986           |

Step 1: 7 Inputs ([M_o, L, f_y, d, ConUP, ReUP] – 1 Output (CBM))

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 10,419                 | 10,919       | 2.96E-04 | 0.984           |
| 2   | 50,000  | 50     | 50      | 50,000         | 7,365                  | 7,865        | 2.44E-04 | 0.986           |

Step 2: 8 Inputs ([f_y, d]^{(a)}, CBM^{(b)}, M_o, L, f_y, ConUP, ReUP) – 1 Output (b)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 2   | 50,000  | 50     | 50      | 50,000         | 7,365                  | 7,865        | 2.44E-04 | 0.986           |

Step 3: 9 Inputs ([M_o, f_y, d]^{(a)}, (CBM, b)_{(b)}, L, ConUP, ReUP) – 1 Output (p)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 6,064                  | 6,264        | 1.33E-03 | 0.985           |

Step 4: 10 Inputs ([f_y, d]^{(a)}, (CBM, b, p)_{(b)}, M_o, L, d, ConUP, ReUP) – 1 Output (p)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 37,046                 | 37,546       | 1.17E-06 | 0.999           |

(a) Selected feature indexes excluding output parameters.
(b) Feature indexes gained from sequence.

(c) CTS (based on more features indexes than recommended by NCA)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 10,419                 | 10,919       | 2.96E-04 | 0.984           |
| 2   | 50,000  | 50     | 50      | 50,000         | 7,365                  | 7,865        | 2.44E-04 | 0.986           |

(d) CRS (based on more features indexes than recommended by NCA)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 10,419                 | 10,919       | 2.96E-04 | 0.984           |
| 2   | 50,000  | 50     | 50      | 50,000         | 7,365                  | 7,865        | 2.44E-04 | 0.986           |

Step 1: 7 Inputs ([M_o, f_y, d]^{(a)}, L, f_y, ConUP, ReUP) – 1 Output (CBM)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 10,419                 | 10,919       | 2.96E-04 | 0.984           |
| 2   | 50,000  | 50     | 50      | 50,000         | 7,365                  | 7,865        | 2.44E-04 | 0.986           |

Step 2: 8 Inputs ([f_y, d]^{(a)}, (CBM, b)_{(b)}, L, ConUP, ReUP) – 1 Output (b)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 6,064                  | 6,264        | 1.33E-03 | 0.985           |

Step 3: 9 Inputs ([f_y, d]^{(a)}, (CBM, b, p)_{(b)}, M_o, L, d, ConUP, ReUP) – 1 Output (p)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 6,952                  | 7,152        | 4.29E-04 | 0.991           |

Step 4: 10 Inputs ([f_y, d]^{(a)}, (CBM, b, p)_{(b)}, M_o, L, d, ConUP, ReUP) – 1 Output (p)

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE  | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|--------------|------|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 37,046                 | 37,546       | 1.17E-06 | 0.999           |

(a) Selected feature indexes excluding output parameters.
(b) Feature indexes gained from sequence.

is suitable for ranking features in distance-based supervised models as shown in MathWorks (2020a). In embedded methods, feature importance is obtained based on the results of training (Guyon and Elisseeff 2003). Feature scores are obtained after training, whereas features need to be selected before training. Other methods applicable for classification problems are available. Only F-test, NCA and RRelieff are usable for preliminary features selection. The NCA method gives much more robust recommendations for selecting features compared with the F-test and RRelieff methods.

In Tables 2(a) and (b), the feature selection scores that affect each other among 11 design parameters (seven input parameters; M_o, L, f_y, d, ConUP, and ReUP and four output parameters; b, p_o, p_c, and CBM) are identified based on the F-test and NCA, finding dominant features in predicting the design parameters including M_o, tensile and compressive rebar ratios (p_o and p_c), and d. These tables also compare the feature scores for predicting outputs such as tensile and...
compressive rebar ratios ($\rho_t$ and $\rho_c$) using the 20,000 datasets based on F-test and NCA. However, some differences are observed, thus necessitating the exploration of the most appropriate method for the considered structural datasets. To improve the training accuracy, only input features that have a substantial impact on predictions of output parameters should be used. Feature selection can aid in better training to create a mathematical model of structural data that will be useful for engineers who seek structural designs based on reverse analysis. In this study, features that have maximum influence on the predictions of output parameters based on NCA were found useful and used to establish input feature indexes to predict the output parameters.

5. Training methods; designs based on TED, PTM, CTS, and CRS

A doubly reinforced concrete beam with seven input parameters ($M_n$, $L$, $f_p$, $f_y$, $d$, $ConUP$, and $ReUP$) was designed for four outputs ($b$, $\rho_t$, $\rho_c$, and CBM). Such

Table 4. Comparison of reverse design based on TED, PTM, CTS, and CRS for $M_n$ of 500 kN·m.

(a) TED

| No. | Parameter | Training results | ANN (TED) results | Autobeam check results | Error (%) |
|-----|-----------|------------------|--------------------|------------------------|-----------|
| 1   | $M_n$ (kN·m) | 500.0 | 684.7 | -36.95 |
| 2   | $L$ (mm) | 10000 | 10000 | 0.00 |
| 3   | $f_p$ (MPa) | 600 | 600 | 0.00 |
| 4   | $f_y$ (MPa) | 50 | 50 | 0.00 |
| 5   | $d$ (mm) | 500 | 500 | 0.00 |
| 6   | $ConUP$ (KRW/m³) | 151007 | 151007 | 0.00 |
| 7   | $ReUP$ (KRW/kg) | 988 | 988 | 0.00 |
| 8   | CBM (KRW/m) | 65454.67 | 68333.22 | -4.40 |
| 9   | $b$ (mm) | (30-35) 3,718 epochs; MSE=1.03E-2; $R=0.876$ | 476.61 | 0.0102 | 0.00 |
| 10  | $\rho_t$ | 0.0028 | 0.0028 | 0.00 |
| 11  | $\rho_c$ | 0.0028 | 0.0028 | 0.00 |

7 Inputs of Artificial Neural Network (ANN)
4 Outputs of Artificial Neural Network (ANN)
9 Forward inputs of Autobeam check
2 Forward outputs of Autobeam check

(b) PTM

| No. | Parameter | Training results | ANN (PTM) results | Autobeam check results | Error (%) |
|-----|-----------|------------------|--------------------|------------------------|-----------|
| 1   | $M_n$ (kN·m) | 500.0 | 523.7 | -4.73 |
| 2   | $L$ (mm) | 10000 | 10000 | 0.00 |
| 3   | $f_p$ (MPa) | 600 | 600 | 0.00 |
| 4   | $f_y$ (MPa) | 50 | 50 | 0.00 |
| 5   | $d$ (mm) | 500 | 500 | 0.00 |
| 6   | $ConUP$ (KRW/m³) | 151007 | 151007 | 0.00 |
| 7   | $ReUP$ (KRW/kg) | 988 | 988 | 0.00 |
| 8   | CBM (KRW/m) | (30-35) 10,419 epochs; MSE=2.96E-4; $R=0.986$ | 61175.66 | 62335.42 | -1.90 |
| 9   | $b$ (mm) | (30-35) 1,502 epochs; MSE=2.59E-4; $R=0.961$ | 470.86 | 470.86 | 0.00 |
| 10  | $\rho_t$ | (30-35) 2,761 epochs; MSE=7.19E-3; $R=0.818$ | 0.0078 | 0.0078 | 0.00 |
| 11  | $\rho_c$ | (30-35) 2,632 epochs; MSE=9.76E-3; $R=0.67$ | 0.0034 | 0.0034 | 0.00 |

7 Inputs of Artificial Neural Network (ANN)
4 Outputs of Artificial Neural Network (ANN)
1 Reverse input of Artificial Neural Network (ANN)
9 Forward inputs of Autobeam check
2 Forward outputs of Autobeam check

(Continued)
a design is defined as a reverse design because the targeted nominal moment capacity \((M_n)\) is pre-assigned on the input side to design beam parameters such as beam width \((b)\), both tensile \((\rho_t)\) and compressive \((\rho_c)\) rebar ratios, and \(CBM\) on the output side. In addition, CTS and CRS training methods based on ANNs with shallow and deep layers are proposed to design doubly reinforced concrete beams.

### 5.1. TED

Training ANNs simultaneously on multiple inputs and outputs, based on TED, is one of the simplest methods. This is because all input parameters are mapped to all output parameters during the training. The seven input parameters \((M_n, L, f_y, f_c, d, ConUP, ReUP)\) listed in Table 2(a) were selected as input feature indexes to predict the four outputs \((b, \rho_t, \rho_c, CBM)\) for designing doubly reinforced beams. The feature selection scores listed in Table 2(b), obtained using the NCA method, demonstrate how strongly the prediction of the output parameters is influenced by the selected input feature indexes. As shown in Table 2(b), the input feature indexes such as \(M_n\) (6.06), \(f_y\) (2.98), \(f_c\) (3.37), and \(d\) (8.55) influence the prediction of the tensile rebar ratio \((\rho_t)\) more substantially than...
Table 5. Comparison of reverse design based on TED and CTS for $M_n$ of 3000 and 4000 kN-m.

(a) For $M_n$ = 3000 kN-m

(1) TED

| No. | Parameter   | Training results | ANN (TED) results | Autobeam check | Error (%) |
|-----|-------------|------------------|-------------------|----------------|----------|
| 1   | $M_n$ (kN-m) | 3000.0           | 3000.0            | 3046.8         | -1.56%   |
| 2   | L (mm)      | 10000            | 10000             | 10000          | 0.00%    |
| 3   | $f_y$ (MPa) | 600              | 600               | 600            | 0.00%    |
| 4   | $f'_{y}$ (MPa) | 50              | 50                | 50             | 0.00%    |
| 5   | d (mm)      | 900              | 900               | 900            | 0.00%    |
| 6   | ConUP (KRW/m$^3$) | 151007        | 151007            | 151007         | 0.00%    |
| 7   | ReUP (KRW/kg) | 988             | 988               | 988            | 0.00%    |

7 Inputs of Artificial Neural Network (ANN)
4 Outputs of Artificial Neural Network (ANN)
1 Reverse input of Artificial Neural Network (ANN)
2 Forward inputs of Autobeam check
2 Forward outputs of Autobeam check

(2) CTS

| No. | Parameter   | Training results | ANN (CTS) results | Autobeam check | Error (%) |
|-----|-------------|------------------|-------------------|----------------|----------|
| 1   | $M_n$ (kN-m) | 3000.0           | 3046.8            | 3046.8         | -1.56%   |
| 2   | L (mm)      | 10000            | 10000             | 10000          | 0.00%    |
| 3   | $f_y$ (MPa) | 600              | 600               | 600            | 0.00%    |
| 4   | $f'_{y}$ (MPa) | 50              | 50                | 50             | 0.00%    |
| 5   | d (mm)      | 900              | 900               | 900            | 0.00%    |
| 6   | ConUP (KRW/m$^3$) | 151007        | 151007            | 151007         | 0.00%    |
| 7   | ReUP (KRW/kg) | 988             | 988               | 988            | 0.00%    |

7 Inputs of Artificial Neural Network (ANN)
4 Outputs of Artificial Neural Network (ANN)
1 Reverse input of Artificial Neural Network (ANN)
2 Forward inputs of Autobeam check
2 Forward outputs of Autobeam check

CTS sequence: CBM $\Rightarrow$ b $\Rightarrow$ $\rho$ $\Rightarrow$ $\rho$

| No. | Parameter   | Training results | ANN results | Error (%) |
|-----|-------------|------------------|-------------|-----------|
| 8   | CBM (KRW/m) | 146078.2         | 146300.0    | 0.03%     |
| 9   | b (mm)      | (30-35) 10,419 epochs; | 529.48     | 529.48    | 0.00%     |
| 10  | $\rho$      | (30-35) 5,358 epochs; | 0.0127     | 0.0127    | 0.00%     |
| 11  | $\rho$      | (30-35) 1,717 epochs; | 0.0029     | 0.0029    | 0.00%     |

Errors of predictions based on CTS are $9.89E-3 \text{ and } 1.03E-2 \text{ irrespective of the types of layers and neurons implemented.}$

L (5.14E-322), ConUP (3.36E-21), and ReUP (1.77E-230). CBM mostly influences the prediction of rebar ratio ($\rho$), as indicated by the feature selection score of 12.24. However, in TED, CBM cannot be used as a feature index to predict the rebar ratio ($\rho$) because both CBM and $\rho$ are located on the output side, and thus, CBM cannot be mapped to $\rho$ during training. Similarly, the parameters placed in the network output cannot be used as input feature indexes to train ANNs on the other output parameters as can be seen for TED, as shown in Table 3(a). Substantially lowered training accuracies are inevitable when CBM, which has the maximum influence on $\rho$, cannot be used as a feature index to train network on $\rho$. It is also difficult to debug the ANN training with deep layers when the networks are trained based on TED, because big computing facilities are needed for TED. As shown in Table 3(a), weak training accuracies are observed when TED is implemented to map the seven input parameters to the four output parameters for the design of doubly reinforced beams. Training accuracies can be judged when ANN-trained models are used to design for given arbitrary input vectors. In addition, weak training accuracies (MSE) are found between $9.89E-3 \text{ and } 1.03E-2 \text{ irrespective of the types of layers and neurons implemented.}$ Errors of predictions based on
TED-trained model are significant (greater than 10%) when they are compared with results calculated from Autobeam software. Verifications of trained models based on TED are presented in Tables 4(a), 5(a)-1, and 5(b)-1. These accuracies are not sufficient to design doubly reinforced beams. Alternatively, modified methods, such as a CTS and CRS, should be implemented based on NCA suggestions to decrease the MSE and increase the R value as shown in Tables 3(c,d). Accordingly, design accuracies are improved significantly based on CTS- and CRS-trained models as shown in Tables 4(c,d), 5(a)-2, and 5(b)-2. These methods allow the design parameters appearing on the output side to be used as input feature indexes to predict the other outputs, allowing all outputs to be simultaneously implemented as feature indexes to predict parameters located on the output side. The sequence of feature indexes used to train the networks on the output parameters can be selected based on the feature scores. Better accuracies (MSE) [between 4.29E-4 and 1.17E-6; Table 3(c,d)] are obtained using CTS and CRS than those (MSE) calculated using TED [between 9.89E-3 and 1.03E-2; Table 3(a)].
5.2. PTM

The training accuracies (MSE) degrade when ANNs are trained on TED when the variables in the network output are required for predicting other outputs. The PTM separates the outputs into groups to train them simultaneously. Although this method requires lesser training time than TED, the feature indexes appearing in the network output still cannot be used as input feature indexes to predict other outputs at the same time. The training accuracy [MSE, 7.19E-3; Table 3(b)] obtained for the tensile rebar ratio (ρ_t) based on PTM is lower than that obtained based on CTS [MSE, 1.24E-4; Table 3(c)].

5.3. CTS

5.3.1. Motivations for CTS

The input feature indexes that sufficiently affect the predictions during training are identified based on the NCA feature selection, and then implemented. CTS has been proposed to address the drawbacks of low training accuracies of the group training method (PTM). In
CTS, all design parameters recommended by NCA feature selection are implemented as feature indexes regardless of whether they belong to the input or output side. This method allows the design parameters recommended by NCA feature selection appearing on the output side to be simultaneously used as the input feature indexes for predicting the other outputs in a chained fashion, improving the training accuracies. The trained output can be used as an input feature index for the other output prediction. The output parameters obtained after training and designing are used as input feature indexes for another training in a chained fashion, thereby restoring the training accuracies substantially. In CTS, a sequence of feature indexes is intuitively determined for training networks on the output parameters. If the network trainings follow a chained fashion based on the NCA feature scores, the networks can be easily debugged while being trained rapidly and accurately. NCA feature selection suggests which feature indexes should be included in the input.

5.3.2. Steps for CTS

Figure 4 presents the chained procedure followed for training the ANNs on the seven inputs and four outputs. Each output is predicted based on the feature indexes selected by the feature scores, which elicit the features that have an increased effect on the current outputs. The feature selection shown in Table 2(b) and Figure 4 recommends six input feature indexes \((M_{n}, f_{o}, d, b, \rho_{o}, \rho_{c})\) to predict CBM; however, the three parameters \((b, \rho_{o}, \rho_{c})\), indicated by blue circles, cannot be used as input feature indexes to predict CBM because they are also output parameters. Figure 4(a), can be used to train ANNs on CBM. The blue parameters were not used as input feature indexes because they belong to the output side at the same time. The CTS established a sequence of intuitive training steps to maximize the training efficiencies.

In Step 1, only three input feature indexes \((M_{n}, f_{o}, d)\), indicated by magenta circles in Figure 4 and Table 3(c), can be used to train ANNs on CBM. Three blue parameters \((b, \rho_{o}, \rho_{c})\) were not used as input feature indexes because they belong to the output side at the same time. The CTS established a sequence of intuitive training steps to maximize the training efficiencies.

In Step 2, the feature selection shown in Table 2(b) and Figure 4 recommends five input feature indexes \(f_{o}, d, \rho_{o}, \rho_{c}, \) and CBM to predict the beam width \(b\); however, two of these parameters \((\rho_{o} \) and \( \rho_{c} \)), indicated in blue circles, cannot yet be used as input feature indexes to predict a beam width \(b\) because they still belong to output parameters of the output side after Step 1.

Figure 4, Step 2 includes CBM, which was predicted in Step 1, as one of the required input feature indexes when training ANNs on \(b\), because it achieves a feature score of 16.39 on \(b\), primarily helping to predict \(b\), as shown by the features listed in Table 2(b) and by magenta circles in Figure 4. CBM can now be used as input feature indexes to train ANNs on a beam width \(b\), thus, three input feature indexes \(f_{o}, d, \) and CBM, indicated in magenta circles in Figure 4 were used to train ANNs on a beam width \(b\). In addition to the three input feature indexes \(f_{o}, d, \text{ and CBM}\), three extra input feature indexes \(M_{n}, \text{ConUP, and ReUP}\) were added based on the trial-and-error technique. When the number of inputs used for training ANNs on \(b\) was increased to eight \((M_{n}, f_{o}, f_{p}, f_{y}, d, \text{ CBM, ConUP, and ReUP})\), i.e., including CBM, the training accuracy obtained using CTS was substantially improved (1.33E-3) compared to that obtained by excluding CBM (2.59E-2) using PTM, as shown in Table 3(b,c). The tensile and compressive rebar ratios \((\rho_{o} = 9.02 \text{ and } \rho_{c} = 3.54)\), indicated in blue circles in the Figure 4, cannot be yet used as input feature indexes to train ANNs on \(b\), because they still belong to the output side in Step 2.
However, the rebar ratios ($\rho_t; 9.02$, $\rho_c; 3.54$) as shown in feature selection table: Table 2(b) should be included as input feature indexes for a better prediction of beam width ($b$) as they are the necessary input feature indexes. Table 2(b). To improve the weak training accuracies, the training sequence can be revised as $CBM \Rightarrow b \Rightarrow \rho_c \Rightarrow \rho_t$. In Step 3, the input feature selection shown in Table 2(b) and Figure 4 recommends seven input feature indexes ($M_n, f_n, d, \rho_c, \rho_t, CBM$) to predict the tensile rebar ratio ($\rho_t$); however, the compressive rebar ratio ($\rho_c$), indicated in blue circle, cannot be used as an input feature index. The training accuracy was further enhanced to $MSE$ of $2.13E-4$ compared to the value of $1.33E-3$ shown in Tables 3(c,d) by using CTS when the training sequence based on NCA features was revised to include the tensile rebar ratio ($\rho_t$) for training ANNs.
input feature index to predict the tensile rebar ratio ($\rho_t$) because the compressive rebar ratio ($\rho_c$) also acts as an output parameter. Six input feature indexes ($M_n$, $f_y$, $f'_c$, $d$, $b$, and $CBM$), indicated in magenta circles in Figure 4 can be used to train ANNs on the tensile rebar ratio ($\rho_t$). The CTS established an intuitive training sequence to maximize the training efficiencies. The beam depth ($d$) was used to train ANNs on $\rho_t$ because $\rho_t$ was affected by $d$ (8.55). Step 3 also implements $CBM$ and $b$ predicted in Steps 1 and 2 to train ANNs on $\rho_t$, which was influenced by $CBM$ (12.24) and $b$ (7.43) based on NCA [Table 2(b)]. Note that $L$, $ConUP$, and $ReUP$ found based on a trial-and-error technique were added to the six input feature indexes, making a total of nine inputs ($M_n$, $L$, $f_y$, $f'_c$, $d$, $b$, $CBM$, $ConUP$, $ReUP$, and $b$), to improve the training accuracies. The training accuracies obtained with (CTS; 1.24E-4) and without $CBM$ and $b$ (PTM; 7.19E-3) are compared in Table 3(b,c), where both $MSE$ and $R$ values were improved when $CBM$ and $b$ were in the input side as input feature indexes. $CBM$ (12.24) and $b$ (7.43) based on NCA [Table 2(b)]. Note that $L$, $ConUP$, and $ReUP$ found based on a trial-and-error technique were added to the six input feature indexes, making a total of nine inputs ($M_n$, $L$, $f_y$, $f'_c$, $d$, $b$, $CBM$, $ConUP$, $ReUP$, and $b$), to improve the training accuracies. The training accuracies obtained with (CTS; 1.24E-4) and without $CBM$ and $b$ (PTM; 7.19E-3) are compared in Table 3(b,c), where both $MSE$ and $R$ values were improved when $CBM$ and $b$ were in the input side as input feature indexes. $CBM$ (12.24) and $b$ (7.43) based on NCA [Table 2(b)]. Note that $L$, $ConUP$, and $ReUP$ found based on a trial-and-error technique were added to the six input feature indexes, making a total of nine inputs ($M_n$, $L$, $f_y$, $f'_c$, $d$, $b$, $CBM$, $ConUP$, $ReUP$, and $b$), to improve the training accuracies. The training accuracies obtained with (CTS; 1.24E-4) and without $CBM$ and $b$ (PTM; 7.19E-3) are compared in Table 3(b,c), where both $MSE$ and $R$ values were improved when $CBM$ and $b$ were in the input side as input feature indexes.

### Table 6. (Continued)

(c) CTS for $d=700$ mm

| No. | Parameter | Training results | ANN (CTS) results | Autobeam check | Error (%) |
|-----|-----------|------------------|-------------------|----------------|-----------|
| 1   | $M_n$ (kN/m) | 500.0 | 548.8 | 9.76% |
| 2   | $L$ (mm) | 10000 | 10000 | 0.00% |
| 3   | $f_y$ (MPa) | 600 | 600 | 0.00% |
| 4   | $f'_c$ (MPa) | 50 | 50 | 0.00% |
| 5   | $d$ (mm) | 700 | 700 | 0.00% |
| 6   | $ConUP$ (KRW/m) | 151007 | 151007 | 0.00% |
| 7   | $ReUP$ (KRW/kg) | 988 | 988 | 0.00% |

The CTS sequence: $CBM \Rightarrow b \Rightarrow \rho \Rightarrow \rho$

7 Inputs of Artificial Neural Network (ANN)

4 Outputs of Artificial Neural Network (ANN)

1 Reverse input of Artificial Neural Network (ANN)

9 Forward inputs of Autobeam check

2 Forward outputs of Autobeam check

(d) CRS for $d=500$ mm

| No. | Parameter | Training results | ANN (CRS) results | Autobeam check | Error (%) |
|-----|-----------|------------------|-------------------|----------------|-----------|
| 1   | $M_n$ (kN/m) | 500.0 | 502.3 | 0.46% |
| 2   | $L$ (mm) | 10000 | 10000 | 0.00% |
| 3   | $f_y$ (MPa) | 600 | 600 | 0.00% |
| 4   | $f'_c$ (MPa) | 50 | 50 | 0.00% |
| 5   | $d$ (mm) | 500 | 500 | 0.00% |
| 6   | $ConUP$ (KRW/m) | 151007 | 151007 | 0.00% |
| 7   | $ReUP$ (KRW/kg) | 988 | 988 | 0.00% |

The CRS sequence: $CBM \Rightarrow \rho \Rightarrow b \Rightarrow \rho$

7 Inputs of Artificial Neural Network (ANN)

4 Outputs of Artificial Neural Network (ANN)

1 Reverse input of Artificial Neural Network (ANN)

9 Forward inputs of Autobeam check

2 Forward outputs of Autobeam check

(Continued)
must be included in the input when training ANNs on b [16.39, trained in Step 2, Figure 4] and \( \rho_c \) [12.24, trained in Step 3, Figure 4] according to Table 2(b) because CBM is strongly recommended. Thus, there is only one remaining output (\( \rho_c \)) after Step 3.

In Step 4, all output variables can now be used to train the compressive rebar ratio (\( \rho_c \)). The tensile rebar ratio (\( \rho_t \)) obtained from Step 3 was used as an input feature index to train ANNs on compressive rebar ratio (\( \rho_c \)) (last output to predict), because \( \rho_t \) was mostly influenced by \( \rho_c \) (8.14), as observed in the feature table [Table 2(b), Figure 4]. The feature selection shown in Table 2(b) and Figure 4 recommends three input feature indexes (\( f_y, f'_c, \rho_c \)) to predict \( \rho_c \), which is indicated in magenta circles, as shown in Step 4. As indicated in Table 2(b), the input features selected on \( \rho_c \) are \( f_y \) (5.11), \( f'_c \) (4.59), and \( \rho_t \) (8.14); however, additional seven input feature indexes (\( M_y, L, d, \text{ConUP}, \text{ReUP}, \text{CBM}, \text{b} \)) were added based on a trial-and-error technique, resulting in 10 inputs (\( M_y, L, f_y, f'_c, d, \text{CBM}, \text{ConUP}, \text{ReUP}, b, \rho_t \)) to train ANNs on \( \rho_c \). Training accuracies with (CTS; 1.17E-6) were significantly improved compared to those without CBM and b (PTM; 9.76E-3), as shown in Table 3(b,c) where both

---

**Table 6. (Continued).**

| No. | Parameter | Training results | ANN (CRS) results | Autobeam check | Error (%) |
|-----|-----------|------------------|-------------------|----------------|-----------|
| 1   | \( M_y \) (kN-m) | \( 500.0 \) | \( 511.7 \) | 2.34% |
| 2   | \( L \) (mm) | \( 10000 \) | \( 10000 \) | 0.00% |
| 3   | \( f_y \) (MPa) | \( 600 \) | \( 600 \) | 0.00% |
| 4   | \( f'_c \) (MPa) | \( 50 \) | \( 50 \) | 0.00% |
| 5   | \( d \) (mm) | \( 600 \) | \( 600 \) | 0.00% |
| 6   | \text{ConUP} (KRW/m) | \( 151007 \) | \( 151007 \) | 0.00% |
| 7   | \text{ReUP} (KRW/kg) | \( 988 \) | \( 988 \) | 0.00% |
| 8   | \text{CBM} (KRW/m) | \( 30-35 \) 10,419 epochs; MSE = 2.96E-4; R = 0.984 | \( 59988.82 \) | 0.02% |
| 9   | \( b \) (mm) | \( 30-35 \) 9,891 epochs; MSE = 2.13E-4; R = 0.997 | \( 433.28 \) | 0.00% |
| 10  | \( \rho_t \) | \( 30-35 \) 6,952 epochs; MSE = 4.29E-4; R = 0.991 | \( 0.0057 \) | 0.00% |
| 11  | \( \rho_c \) | \( 30-35 \) 37,046 epochs; MSE = 1.17E-6; R = 0.999 | \( 0.0015 \) | 0.00% |

CRS sequence: CBM \( \Rightarrow \rho_c \Rightarrow b \Rightarrow \rho_t \)

---

**Table 6. (Continued).**

| No. | Parameter | Training results | ANN (CRS) results | Autobeam check | Error (%) |
|-----|-----------|------------------|-------------------|----------------|-----------|
| 1   | \( M_y \) (kN-m) | \( 500.0 \) | \( 511.7 \) | -8.82% |
| 2   | \( L \) (mm) | \( 10000 \) | \( 10000 \) | 0.00% |
| 3   | \( f_y \) (MPa) | \( 600 \) | \( 600 \) | 0.00% |
| 4   | \( f'_c \) (MPa) | \( 50 \) | \( 50 \) | 0.00% |
| 5   | \( d \) (mm) | \( 700 \) | \( 700 \) | 0.00% |
| 6   | \text{ConUP} (KRW/m) | \( 30-35 \) 10,419 epochs; MSE = 2.96E-4; R = 0.984 | \( 58713.75 \) | -0.01% |
| 7   | \text{ReUP} (KRW/kg) | \( 988 \) | \( 988 \) | 0.00% |
| 8   | \text{CBM} (KRW/m) | \( 30-35 \) 9,891 epochs; MSE = 2.13E-4; R = 0.997 | \( 377.52 \) | 0.00% |
| 9   | \( b \) (mm) | \( 30-35 \) 6,952 epochs; MSE = 4.29E-4; R = 0.991 | \( 0.0051 \) | 0.00% |
| 10  | \( \rho_t \) | \( 30-35 \) 37,046 epochs; MSE = 1.17E-6; R = 0.999 | \( 0.0015 \) | 0.00% |

CRS sequence: CBM \( \Rightarrow \rho_c \Rightarrow b \Rightarrow \rho_t \)
Table 7. Chaining procedure with revised sequence (CRS) based on features recommended by NCA only.

(a) Training on first parameter (CBM) based on only selected features

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|---------------|-----|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 4,029                  | 4,529         | 3.21E-04 | 0.982           |
| 2   | 50,000  | 50     | 50      | 50,000         | 4,052                  | 4,552         | 3.46E-04 | 0.98            |

[a] Selected feature indexes excluding output parameters

(b) Training on second parameter ($\rho_t$) based on only selected features

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|---------------|-----|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 5,583                  | 5,783         | 8.25E-04 | 0.98            |

[a] Selected feature indexes excluding output parameters

[c] Training on third parameter ($r$) based on only selected features

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|---------------|-----|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 3,251                  | 3,451         | 3.48E-03 | 0.996           |

[b] Feature indexes gained from sequence

(d) Training on fourth parameter ($\rho_t$) based on selected features

| No. | Dataset | Layers | Neurons | Required Epoch | Best Epoch for Training | Stopped Epoch | MSE | R at Best Epoch |
|-----|---------|--------|---------|----------------|------------------------|---------------|-----|-----------------|
| 1   | 20,000  | 30     | 35      | 50,000         | 1,509                  | 1,709         | 3.94E-03 | 0.883           |

[a] Selected feature indexes excluding output parameters.

[a] Selected feature indexes excluding output parameters.

[b] Feature indexes gained from sequence.

[a] Selected feature indexes excluding output parameters.

[b] Feature indexes gained from sequence.
MSE and $R$ values show improvement with $CBM$, $b$, and $\rho_t$ obtained from Steps 1, 2, and 3, respectively, on the input side. Tables 3(a,c) compare the training accuracies between TED and CTS based on NCA feature selection. Figure 4 summarizes the four steps followed for CTS training of ANNs on the four outputs ($CBM$, $b$, $\rho_t$, and $\rho_c$) based on the input feature scores listed in Table 2(b) and Figure 4. The prediction sequence for CTS is intuitively determined as follows (Figure 4):

1. Step 1 (Figure 4): ANNs are trained on $CBM$ based on CTS, and the remaining three variables ($b$, $\rho_t$, $\rho_c$) on the output side are predicted after Step 1.

2. Step 2 (Figure 4): ANNs are trained on $b$ ($b$ is affected by $CBM$ (16.39)) based on CTS after $CBM$ is predicted in Step 1. The remaining two outputs ($\rho_t$, $\rho_c$) remained are predicted after Step 2.

Figure 7. Verification charts of $M_n$ and $CBM$. 
Step 3 (Figure 4): The training of ANNs on $\rho_t$ based on CTS, is influenced by CBM (12.24) and $b$ (7.43) determined in Steps 1 and 2. Thus, only one output ($\rho_c$) remains after Step 3.

Step 4 (Figure 4): In CTS, $\rho_t$ (8.14) is used as an input feature index for $\rho_c$.

5.4. CRS

5.4.1. Motivation for CRS

In CRS, networks can be trained with a revised training sequence to improve the training accuracy of CTS. Better training accuracies can be obtained with a well-determined training sequence based on the obtained feature scores. In a chained fashion, parameters placed on the output side are used as input feature indexes for the other output predictions at the same time. CRS allows the design parameters appearing on the output side to be simultaneously used as input feature indexes of the other output variables. Often, the training quality achieved using only input feature indexes suggested by NCA are not as good as that achieved by including extra input features. Feature selection provides information on features that must be included as
input feature indexes that have substantial impact on the other variables, however, extra features can be added to increase the training accuracy. A trial-and-error technique must be used to identify the extra input feature indexes for achieving better training accuracies and output prediction.

### 5.4.2. Revised sequence based on PTM and CTS

Figure 5 and Table 3(d) elicit the determination of a sequence for a chaining procedure when training networks with CRS. As shown in Table 2(b), the tensile rebar ratio ($\rho_t$) and beam width ($b$) are influenced by CBM with feature scores of 12.24 and 16.39, respectively, requiring CBM to be used as a feature index for both $b$ and $\rho_t$. CBM is, thus, recommended to be trained in Step 1. In the CTS shown in Figure 4, the beam width ($b$) design was determined prior to determining the tensile rebar ratio ($\rho_t$), and the training sequence of $b$ and $\rho_t$ was reversed for CRS, based on the input features indicated in the magenta circles.
### Table 8. Adjusting beam depth to minimize errors based on TED, CTS and CRS for targeted nominal moment strength ($M_n = 500$ kN·m).

(a) Error of $M_n$ and CBM of preassigned $M_n = 500$ kN·m based on TED

| Beam depth $d$ | $M_n$ (kN-m) | Error (%) | CBM (KRW/m) | Error (%) |
|---------------|--------------|-----------|-------------|-----------|
| 200           | 500.00       | 412.02    | 17.60       | 48,434.67 | 81,148.56 | -68.10 |
| 250           | 500.00       | 561.29    | -12.26      | 61,368.19 | 89,697.61 | -46.16 |
| 300           | 500.00       | 662.43    | -32.49      | 68,402.28 | 90,642.67 | -32.51 |
| 350           | 500.00       | 689.12    | -37.82      | 64,179.97 | 84,109.96 | -31.05 |
| 400           | 500.00       | 670.69    | -35.22      | 60,036.71 | 75,890.67 | -26.41 |
| 450           | 500.00       | 672.22    | -34.44      | 61,745.82 | 70,725.66 | -14.54 |
| 500           | 500.00       | 684.74    | -36.95      | 65,454.67 | 68,333.22 | -4.40  |
| 550           | 500.00       | 701.70    | -40.34      | 67,965.52 | 67,139.93 | 1.21   |
| 600           | 500.00       | 713.32    | -42.66      | 68,647.45 | 66,138.62 | 3.63   |
| 650           | 500.00       | 716.65    | -43.33      | 68,024.53 | 65,028.26 | 4.40   |
| 700           | 500.00       | 713.82    | -42.76      | 66,789.75 | 63,745.64 | 4.56   |
| 750           | 500.00       | 709.12    | -41.82      | 65,484.47 | 62,486.85 | 4.61   |
| 800           | 500.00       | 706.91    | -41.38      | 64,471.10 | 61,042.94 | 4.76   |

(b) Error of $M_n$ and CBM of preassigned $M_n = 500$ kN·m based on CTS

| Beam depth $d$ | $M_n$ (kN-m) | Error (%) | CBM (KRW/m) | Error (%) |
|---------------|--------------|-----------|-------------|-----------|
| 200           | 500.00       | 406.89    | 18.62       | 78,660.01 | 78,305.38 | 0.45   |
| 250           | 500.00       | 451.25    | 9.75        | 71,032.72 | 70,994.57 | 0.05   |
| 300           | 500.00       | 475.86    | 4.83        | 66,622.12 | 66,650.40 | -0.04  |
| 350           | 500.00       | 485.03    | 2.99        | 64,144.50 | 64,159.46 | -0.02  |
| 400           | 500.00       | 486.46    | 2.71        | 62,728.22 | 62,718.98 | 0.01   |
| 450           | 500.00       | 488.08    | 2.38        | 61,838.38 | 61,814.07 | 0.04   |
| 500           | 500.00       | 494.13    | 1.17        | 61,175.66 | 61,146.79 | 0.05   |
| 550           | 500.00       | 504.64    | -0.93       | 60,584.29 | 60,557.65 | 0.04   |
| 600           | 500.00       | 517.79    | -3.56       | 59,988.82 | 59,976.45 | 0.03   |
| 650           | 500.00       | 532.39    | -6.48       | 59,361.33 | 59,348.55 | 0.02   |
| 700           | 500.00       | 548.82    | -9.76       | 58,713.75 | 58,709.79 | 0.01   |
| 750           | 500.00       | 568.49    | -13.70      | 58,096.18 | 58,103.88 | -0.01  |
| 800           | 500.00       | 592.60    | -18.52      | 57,614.49 | 57,629.49 | -0.03  |

(c) Error of $M_n$ and CBM of preassigned $M_n = 500$ kN·m based on CRS

| Beam depth $d$ | $M_n$ (kN-m) | Error (%) | CBM (KRW/m) | Error (%) |
|---------------|--------------|-----------|-------------|-----------|
| 200           | 500.00       | 381.92    | 23.62       | 78,660.01 | 77,271.72 | 1.76   |
| 250           | 500.00       | 459.94    | 8.03        | 71,032.72 | 70,911.44 | 0.17   |
| 300           | 500.00       | 496.26    | 0.75        | 66,622.12 | 66,662.17 | -0.06  |
| 350           | 500.00       | 497.14    | 0.57        | 64,144.50 | 64,163.23 | -0.03  |
| 400           | 500.00       | 499.01    | 0.20        | 62,728.22 | 62,719.87 | 0.01   |
| 450           | 500.00       | 501.73    | -0.35       | 61,838.38 | 61,815.75 | 0.04   |
| 500           | 500.00       | 502.30    | -0.46       | 61,175.66 | 61,149.53 | 0.04   |
| 550           | 500.00       | 504.38    | -0.88       | 60,584.29 | 60,562.49 | 0.04   |
| 600           | 500.00       | 511.68    | -2.34       | 59,988.82 | 59,975.85 | 0.02   |
| 650           | 500.00       | 525.18    | -5.04       | 59,361.33 | 59,359.35 | 0.00   |
| 700           | 500.00       | 544.10    | -8.82       | 58,713.75 | 58,722.44 | -0.01  |
| 750           | 500.00       | 567.42    | -13.48      | 58,096.18 | 58,117.60 | -0.02  |
| 800           | 500.00       | 594.31    | -18.86      | 57,614.49 | 57,644.22 | -0.05  |
shown in Figure 5. The blue parameters were not used as input feature indexes because they also belong to the output side. As shown in Table 2(b), tensile rebar ratio ($\rho_b$) is influenced by beam width (b) with a feature score of 7.43, whereas b is influenced by $\rho_b$ with a feature score of 9.02; thus, $\rho_b$ should be trained before b, as shown in Figure 5. As shown in Figure 5, the tensile rebar ratio ($\rho_b$) is used as an input feature index for b. From the design perspective, the sequence of training networks matters because the $\rho_b$ design should be determined prior to the b design. Note that the training results of ANN on b were not good ($MSE = 1.33E-3; R = 0.985$) when $\rho_b$, which plays an important role in predicting b (9.02), was not used as input feature indexes for b, as shown in Table 2(b) and Figure 4. When a sequence of network training on b was revised to include $\rho_b$ as an input feature index, the training accuracies on b were improved significantly to an $MSE$ of 2.13E-4 and R value of 0.997 by CRS, as shown in Table 3(d), compared to the previous sequence ($MSE = 1.33E-3; R = 0.985$ by CTS) shown in Table 3(c).

The compressive rebar ratio ($\rho_c$) is influenced by the tensile rebar ratio ($\rho_b$) with a feature score of 8.14; thus, $\rho_b$ should be used as an input feature index for $\rho_c$. In CRS, the output parameters with good training accuracies should be trained first, without the need for using other output parameters as input feature indexes; however, they can be used to predict the next variables. The previously trained variables can be used as input feature indexes of the output variables that need them. Meanwhile, the output parameters with weak training accuracies should be trained last because they need to capture more input feature indexes for improving the accuracy. Tables 3(a–d) compare the training accuracies among TED, CTS, and CRS. CRS was trained with a revised sequence based on NCA feature selection. The chaining procedure used to train networks with revised sequence (CRS) is presented in Figure 5, which verifies the use of training networks based on CTS and CRS, where improved training accuracies are demonstrated for CRS with a revised training sequence.

5.4.3. Steps for CRS
Similar to CTS, in CRS as well, a training sequence can be determined to maximize the training accuracies, where Step 1 trains ANNs on CBM, as shown in Figure 5. Six inputs ($M_n, f_o, f_d, b, \rho_o$, and $\rho_b$) are suggested to be trained on CBM by NCA. However, the three remaining outputs (b, $\rho_o$, and $\rho_b$) indicated in blue circles, cannot be used as input feature indexes because they belong to the four outputs (CBM, b, $\rho_o$, and $\rho_b$) at the same time. Thus, ANNs are trained on CBM using seven inputs ($M_n, L, f_o, f_d, d, ConUP, and ReUP$), where $L, f_o, ConUP, and ReUP$ are added. Three variables (b, $\rho_o$, and $\rho_b$), indicated in blue circles, cannot be used yet as input feature indexes to predict other output variables because they still belong to the output side after Step 1.

In Step 2, seven inputs ($M_n, f_o, f_d, b, \rho_o$, and CBM) are suggested to train ANNs on $\rho_b$ by NCA. The CBM determined in Step 1 can be now used as an input feature index in Step 2 for training ANNs on $\rho_b$ as shown in Figure 5. The featuring score of CBM on $\rho_b$ is 12.24, as shown Table 2, which significantly contributes to the prediction of $\rho_b$. Three input feature indexes (L, ConUP, and ReUP) are added, resulting in a total of 10 input feature indexes. However, only eight inputs ($M_n, L, f_o, f_d, ConUP, ReUP, and CBM$) are used to train ANNs on $\rho_b$ because the remaining two outputs (b and $\rho_b$), indicated in blue circles, cannot be used as input feature indexes. They are part of the other four outputs (CBM, b, $\rho_o$, and $\rho_b$). Thus, beam width (b) and compressive rebar ratio ($\rho_c$, indicated in blue circles), cannot yet be used as input feature indexes to train ANNs on the other output variables because it still belongs to output side after Step 2.

In Step 3, five inputs ($f_o, d, \rho_o, \rho_b$, CBM) are suggested to train ANNs on b by NCA. In this case, the variables CBM (predicted in Step 1) and $\rho_b$ (predicted in Step 2) can now be included as input feature indexes. The feature scores of CBM and $\rho_b$ are 16.39 and 9.02 on b, respectively, as indicated in Table 2(b), and is primarily attributed to the prediction of b, as shown by the features in magenta circles in Figure 5. Five input feature indexes ($M_n, L, f_o, ConUP, and ReUP$) found through trial and error were additionally used as input feature indexes in Step 3 to improve the training accuracies. Thus, a total of 10 input feature indexes can be used; however, one remaining output ($\rho_b$), indicated in blue circle, cannot be used as an input feature index because it is also a part of the four outputs (CBM, b, $\rho_o$, and $\rho_b$). Thus, only one output ($\rho_c$) remains after Step 3.

In Step 4, all output variables can be used to train the compressive rebar ratio ($\rho_c$) after Step 3. Three inputs, $f_o$ (5.11), $f_d$ (4.59), and $\rho_b$ (8.14), indicated in Table 3(b) were suggested to train ANNs on compressive rebar ratio ($\rho_c$) by NCA. The tensile rebar ratio ($\rho_b$) trained in Step 2 can be used as the input feature index to train ANNs on $\rho_o$ (last output to predict). The variable $\rho_b$ is mostly influenced by $\rho_b$ (8.14), as can be seen in the feature table [Table 2(b)]. The number of inputs required to train ANNs on $\rho_b$ becomes ten ($M_n, L, f_o, f_d, d, ConUP, ReUP, CBM, b$, and $\rho_b$) when seven feature indexes ($M_n, L, d, ConUP, ReUP, CBM$, and b), i.e., including CBM (trained in Step 1) and b (trained in Step 3), are added to enhance training accuracies. Notably, all three outputs (CBM, b, and $\rho_b$) can be used to train ANNs on the last output $\rho_b$. The training accuracies of $\rho_b$ (1.17E-6) based on CTS [Table 3(c)] and CRS [Table 3(d)] were significantly improved from 9.76E-3 obtained based on PTM [Table 3(b)] without considering CBM (Step 1), b (Step 2), and $\rho_b$ (Step 3). The MSE (1.33E-3) of
b weakened using CTS training ANN on b without \( \rho_c \) having a feature score of 9.02 on b [Tables 2(b) and 3(c)]. However, the MSE (2.13E-4) of b based on CRS improved when the training sequence included \( \rho_c \) by reversing training sequence, as shown in Step 3 of CRS [Tables 2(b) and 3(d)]. The MSE obtained when training ANN on \( \rho_c \) without b is 4.29E-4 based on CRS, as shown in Table 3(d), whereas the that of \( \rho_c \) obtained by training ANN on \( \rho_c \) with b increased to 1.24E-4 based on CRS, as shown in Table 3(c).

In Figure 5, following are the four steps summarized for CRS training on four outputs (CBM, \( b, \rho_b, \rho_c \)) based on the feature scores [Table 2(b) and Figure 5]:

1. Step 1: ANNs are trained on CBM based on CRS, and thus, three variables \( (b, \rho_b, \rho_c) \) remain on the output side from four outputs \( (CBM, b, \rho_b, \rho_c) \) after Step 1.
2. Step 2: ANNs are trained on \( \rho_c \) [\( \rho_c \) is affected by CBM (12.24)] based on CRS after CBM is predicted in Step 1. Thus, two outputs \( (b, \rho_b) \) remain on the output side after Step 2.
3. Step 3: The training of ANN on b is influenced by CBM (16.39) and \( \rho_b \) (9.02) based on CRS. CBM and \( \rho_b \) determined in Steps 1 and 2 are implemented for training ANN on b. Thus, there is only one remaining output \( \rho_c \) after Step 3.
4. Step 4: In CRS, \( \rho_c \) (8.14) is used as an input feature index for \( \rho_c \).

The sequence for CRS training on the four outputs (CBM, \( b, \rho_b, \rho_c \)) based on the feature scores [Figure 5] is summarized in Figure 6. The input feature indexes were selected based on the feature scores suggested by NCA by excluding features belonging to the output side at the same time. The feature indexes gained from the sequence and extra features selected to train the output variables are presented. The selection sequence follows Steps 1 to 4. The quality of training ANN on b based on CRS [MSE = 2.13E-4; \( R = 0.997 \), Table 3(d)] was better than that based on CTS [MSE = 1.33E-3; \( R = 0.985 \), Table 3(c)], which indicates that CRS shows better improvement in the training accuracies than CTS by revising the training sequence based on the feature selection scores listed in Table 2(b).

CRS [Tables 6(d–f)] yielded slightly better design accuracies than CTS [Table 6(a–c)]. Moreover, CRS enhanced the design accuracies when the training sequence was revised by training the tensile rebar ratio \( \rho_c \) prior to the beam width \( b \).

### 5.4.4. CRS based on features recommended by NCA only vs. with extra features

Figure 6 illustrates a CRS based on Figure 5, showing how a sequence of feature indexes is selected based on NCA’s recommendation. These features appearing on the output side cannot be used as input feature indexes for other output variables before they are trained to become available.

As shown in Table 7(c), the training accuracy of b is 3.48E-04 based on the four input feature indexes \( (f'_c, d, \rho_c, CBM) \) according to the CRS recommended by NCA, as indicated in magenta parentheses in Table 7(c). The compressive rebar ratio \( \rho_c \), indicated in the blue circle as shown in Figure 5, is excluded from the prediction of b because it also acts as an output parameter. In Table 3(d), the training accuracy of b increases from 3.48E-04 to 2.13E-04 when nine input feature indexes \( (M_a, L, f_b, f_c, d, CBM, ConUP, ReUP, \rho_b) \) including features determined through trial and error are implemented as input feature indexes based on CRS. The training quality achieved by only using the inputs suggested by NCA is not as good as that achieved by including extra input feature indexes. In Table 7(b), the training accuracy of \( \rho_c \) is 8.25E-04 by using only the five input feature indexes \( (M_a, L, f_b, f_c, d, CBM) \) based on the CRS recommended by NCA, indicated in magenta circles in Figure 5. Beam width \( b \) and compressive rebar ratio \( \rho_c \), indicated in blue circles in Figure 5, are excluded from the prediction of \( \rho_c \) because they also act as output parameters. In Table 3(d), the training accuracy of \( \rho_c \) increases to 4.29E-04 when eight input feature indexes \( (M_a, L, f_b, f_c, d, CBM, ConUP, ReUP) \) including features gained from sequence implemented as input feature indexes. The training quality achieved using only inputs suggested by NCA is not as good as that achieved by including extra input feature indexes. This is because although the NCA feature selection method suggests the inputs to be included, it does not recommend the extra input feature indexes required to enhance the training accuracies.

### 5.4.5. Design verifications based on training methods

The training accuracies of CRS shown in Table 3(d) are compared with those of TED shown in Table 3(a) and CTS shown in Table 3(c), where the training accuracies are obtained based on additional feature indexes than those recommended by NCA. In Tables 4–6, all networks were trained on seven inputs \( (M_a, L, f_b, f_c, d, ConUP, and ReUP) \) and four input parameters \( (CBM, b, \rho_b, and \rho_c) \), while nominal moment strengths \( (M_a) \) were reversely pre-assigned on the input side, which is difficult to be achieved using conventional design methods. As shown in Table 4, based on 30 layers and 35 neurons, reverse design results based on TED, PTM, CTS, and CRS were compared for a nominal moment strength \( (M_a) \) of 500 kN⋅m reversely pre-assigned on the input side. The design accuracy achieved using TED with the lowest training quality of MSE = 1.03E-2 and \( R = 0.876 \) [Table 3(a)] led to −36.95% of the nominal moment strength \( (M_a) \), and the errors decreased to −4.73%, 1.17%, and −0.46% when PTM, CTS, and CRS were implemented, respectively. As shown in Tables 4 and 5, a substantial improvement in the
design accuracies was observed when errors of $-36.95\%$, $-26.21\%$, and $-26.14\%$ in case of using TED were reduced to $1.17\%$, $-1.56\%$, and $-1.29\%$ in case of using CTS when the targeted nominal moment strengths ($M_n$) of 500, 3000, and 4000 kN-m were reversely pre-assigned on the input side, respectively. CTS and CRS sufficiently improved the quality of training and design for use in engineering practice. Thus, CTS and CRS, which included extra input feature indexes for training, enhanced the design quality. The best training and design resulted from CRS, as shown in Table 4(d). As shown in Tables 4 and 5, the designs obtained by TED, PTM, CTS, and CRS were presented for $M_n$ values of 500, 3000, and 4000 kN-m, respectively. The resulting $b$ value was unreasonably large when a beam depth ($d$) of 500 mm was assigned to the input side, as shown in Table 4, where the beam depth ($d$) of 500 mm is too shallow to design a pre-assigned targeted moment capacity of 500 kN-m, resulting in beam widths ($b$) of 471–506 mm for TED, PTM, CTS, and CRS, respectively.

Reasonable beam widths ($b$) can be obtained by pre-assigning reasonable beam depths ($d$) to the input side, as shown in Table 5(a), where a beam depth ($d$) of 900 mm is pre-assigned to the input side to design beams for a pre-assigned targeted moment capacity of 3000 kN-m, resulting in beam width ($b$) of 529.5 mm for CTS. Results similar to those shown in Table 5(a) are presented in Table 5(b), where a beam depth ($d$) of 1000 mm is pre-assigned to the input side to design beams for a pre-assigned targeted moment capacity of 4000 kN-m, resulting in a beam width ($b$) of 554.6 mm for CTS.

Design results based on CTS [Tables 6(a–c)] and CRS [Tables 6(d–f)] are presented for a nominal moment strength ($M_n$) of 500 kN-m. CTS and CRS were compared for beam depths of 500, 600, and 700 mm when $M_n$ of 500 kN-m was reversely pre-assigned to the input side. The beam width ($b$) decreases as the beam depth ($d$) increases, as shown in Table 6. Notably, the accuracy of the $M_n$ value based on CRS with $-0.46\%$ error was better than the 1.17% error calculated by CTS, indicating that the training sequences must be carefully determined based on the input features.

The design accuracies based on both CTS and CRS degraded when the beam depths ($d$) outside the dataset range were prescribed on the input side for a nominal moment strength ($M_n$) of 500 kN-m, as shown in Table 6. Suitable design results [1.17% for CTS in Table 6(a) and $-0.46\%$ for CRS in Table 6(d)] were obtained for a beam depth of 500 mm, whereas weak results ($-9.76\%$ and $-8.82\%$, respectively) were obtained for a beam depth of 700 mm. This indicates that the training accuracies of ANNs achieved on the datasets were weak to match and map the beam sections with a beam depth of 700 mm to a nominal moment strength ($M_n$) of 500 kN-m, because a beam depth of 700 mm is too large for a nominal moment strength ($M_n$) of 500 kN-m. ANN better matched the beam sections with a beam depth of 500 mm for $M_n$ of 500 kN-m. Therefore, the input parameters should be carefully selected, especially when variables such as $M_n$ are reversely pre-assigned to the input side. In addition, the reverse input variables must be well-mapped to the output variables based on large structural datasets. Neural networks are not responsible for the errors caused by the mismatched reverse variables assigned to the input side, resulting in input conflicts which can be adjusted before commencing training.

### 5.4.6. Verification charts

As shown in Fig. 7(c–1), errors over 5% associated with $M_n$ start with beam depths smaller than 250 mm or greater than 600 mm, indicating that the range of beam depths ($d$) between 250 to 600 mm should be used for $M_n$ equal to 500 kN-m to keep errors less than 5%. In Figure 7(a)-(1), (b)-(1), and (c)-(1), the errors of $M_n$ corresponding to $d$ of 600 mm (Step 1) were $-42.66$, $3.56$, $-2.34\%$ (Step 3) compared with $M_n$ based on structural mechanics (Autobeam) of 713.3, 517.8, 511.7 kN-m (Step 2) for TED, CTS, and CRS, respectively. Similarly, errors of CBM was also compared with those based on structural mechanics (Autobeam) corresponding to $d$ of 600 mm (Step 1) are obtained as 3.63, 0.03, 0.02 % (Step 3) as shown in Figure 7(a)-(2), (b)-(2), and (c)-(2) for TED, CTS, and CRS, respectively. The design accuracies obtained based on CTS and CRS were acceptable for practical design application. The methods proposed in this study were trained and tested based on MATLAB, R2020b (MathWorks 2020b). The verification was performed by engineering software, Autobeam (Nguyen and Hong 2019). In Figure 7(b)-(1) and (c)-(1), considerable errors occur in the region where datasets were not trained. Figure 7 is tabulated in Table 8.

### 6. Conclusions

This study proposed CRS and CTS to train ANNs with deep layers for a structural design of doubly reinforced concrete beams. A targeted nominal moment capacity ($M_n$) was reversely pre-assigned to the input side to design beam parameters such as beam width ($b$), both tensile ($\rho_s$) and compressive ($\rho_c$) rebar ratios, and CBM on the output side. The proposed method for applications to a design of doubly reinforced concrete beams can be extended to the broad field of structural engineering design. The following conclusions were reached.

1. CRS and CTS performed training on large datasets based on the feature selection scores determined by NCA, to design doubly reinforced concrete beams. Four design outputs ($b$, $\rho_s$, $\rho_c$, CBM) of doubly reinforced concrete beams were determined for given seven input parameters ($M_n$, $L$, $f_y$, $f_{\rho_n}$, $d$, ConUP and ReUP) based on CRS and CTS.
2. CRS determined the training sequence of the input feature indexes based on the
recommendation by NCA; however, the input feature indexes were extended to capture better training accuracies.

(3) Four design outputs \((b, p_0, p_0, CBM)\) of the doubly reinforced concrete beams were determined for the given seven input parameters \((M_r, L, f_p, f_o, d, ConUP\) and \(ReUP\)) based on CRS, CTS, PTM, and TED. As shown in Tables 3(a–c), the designs obtained by CTS, and CRS were verified for \(M_r\) of 500, 3000 and 4000 kN-m, respectively, whereas the best training and design were demonstrated by CRS (Tables 3a)-(4)).

(4) CRS- and CTS-trained ANN models substantially outperformed the PTM- and TED-trained ANN models, with the ability to accurately design doubly reinforced concrete beams with multiple input and output parameters.

**Nomenclature**

| Structural parameters |  |
|-----------------------|-----------------|
| \(M_r\) (kN-m)       | Nominal moment capacity |
| \(L\) (mm)           | Beam span |
| \(d\) (mm)           | Effective beam depth |
| \(b\) (mm)           | Beam width |
| \(p_0\)              | Tensile rebar ratio |
| \(p_o\)              | Compressive rebar ratio |
| \(f_p\) (MPa)        | Yield rebar strength |
| \(f_o\) (MPa)        | Compressive concrete strength |
| \(ConUP\) (KRW/m\(^3\)) | Concrete unit price |
| \(ReUP\) (KRW/kg)   | Rebar unit price |
| \(CBM\) (KRW/m)     | Cost of beam materials |

| Training methods |  |
|------------------|----------------|
| TED              | Training on entire dataset |
| CTS              | Chained training scheme |
| PTM              | Parallel training method |
| CRS              | Chained training scheme with revised sequence |

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**Funding**

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government [MSIT 2019R1A2C2004965].

**Notes on contributors**

**Dr. Won-Kee Hong** is a Professor of Architectural Engineering at Kyung Hee University. Dr. Hong received his Masters and Ph.D. degrees from UCLA, and he worked for Englelklirk and Hart, Inc. (USA), Nihhon Sekkei (Japan) and Samsung Engineering and Construction Company (Korea) before joining Kyung Hee University (Korea). He also has professional engineering licenses from both Korea and the USA. Dr. Hong has more than 30 years of professional experience in structural engineering. His research interests include new approaches to construction technologies based on value engineering with hybrid composite structures. He has provided many useful solutions to issues in current structural design and construction technologies as a result of his research combining structural engineering with construction technologies. He is the author of numerous papers and patents, both in Korea and the USA. Currently, Dr. Hong is developing new connections that can be used with various types of frames, including hybrid steel-concrete precast composite frames, precast frames and steel frames. These connections would contribute to the modular construction of heavy plant structures and buildings as well. He recently published a book titled as “Hybrid Composite Precast Systems: Numerical Investigation to Construction” (Elsevier).

**Tien Dat Pham** received a Master’s degree from Seoul National University of Science and Technology. He is currently enrolled as a Ph.D. candidate in the Department of Architectural Engineering at Kyung Hee University, Republic of Korea. His research interest includes structural analyses using numerical methods considering material plasticity. Furthermore, his studies also involve developing optimal designing methods based on machine learning and genetic algorithms.

**Van Tien Nguyen** is currently enrolled as a Master candidate in the Department of Architectural Engineering at Kyung Hee University, Republic of Korea. His research interests include a design of precast composite structures based on Artificial Neural Network (ANN).

**References**

Abdalla, K. M., and G. E. Stavroulakis. 1995. “A Backpropagation Neural Network Model for Semi-rigid Steel Connections.” *Computer-Aided Civil and Infrastructure Engineering* 10 (2): 77–87. https://doi.org/10.1111/j.1467-8667.1995.tb00271.x .

Azim, I., J. Yang, M. F. Iqbal, M. F. Javed, S. Nazar, F. Wang, and Q. F. Liu. 2020a. “Semi-analytical Model for Compressive Arch Action Capacity of RC Frame Structures.” *Structures, Elsevier* 27: 1231–1245. doi:10.1016/j.istruc.2020.06.011.

Azim, I., J. Yang, M. F. Iqbal, Z. Mahmood, M. F. Javed, F. Wang, and Q. F. Liu. 2021. “Prediction of Catenary Action Capacity of RC Beam-Column Substructures under a Missing Column Scenario Using Evolutionary Algorithm.” *KSCE Journal of Civil Engineering, Springer* 25: 891–905. doi:10.1007/s12205-021-0431-0.

Azim, I., J. Yang, M. F. Javed, M. F. Iqbal, Z. Mahmood, F. Wang, and Q. F. Liu. 2020b. “Prediction Model for Compressive Arch Action Capacity of RC Frame Structures under Column Removal Scenario Using Gene Expression Programming.” *Structures, Elsevier* 25: 212–228. doi:10.1016/j.istruc.2020.02.028.

Azimi-Pour, M., and H. Esfandari-Naddaf. 2018. “ANN and GEP Prediction for Simultaneous Effect of Nano and Micro Silica on the Compressive and Flexural Strength of Cement Mortar.” *Construction and Building Materials* 189: 978–992. doi:10.1016/j.conbuildmat.2018.09.031.

Chen, H. M., K. H. Tsai, G. Z. Qi, J. C. S. Yang, and F. Amini. 1995. “Neural Network for Structure Control.” *Journal of
Computing in Civil Engineering 9 (2): 168–176. doi:10.1061/(ASCE)0887-3801(1995)9:2(168).

Gandomi, A. H., A. H. Alavi, M. R. Mirzahosseini, and F. M. Nejad. 2011. “Nonlinear Genetic-based Models for Prediction of Flow Number of Asphalt Mixtures.” *Journal of Materials in Civil Engineering* 23 (3): 248–263. doi:10.1061/(ASCE)MT.1943-5533.0000154.

Gupta, T., and R. K. Sharma. 2011. “Structural Analysis and Design of Buildings Using Neural Network: A Review.” *International Journal of Engineering and Management Sciences* 2 (4): 216–220.

Guyon, I., and A. Elisseeff. 2003. “An Introduction to Variable and Feature Selection.” *Journal of Machine Learning Research* 3: 1157–1182.

Hong, W. K. 2019. *Hybrid Composite Precast Systems: Numerical Investigation to Construction*, 427–478. Woodhead Publishing, Elsevier.

Iqbal, M. F., Q. F. Liu, I. Azim, X. Zhu, J. Yang, M. F. Javed, and M. Rauf. 2020. “Prediction of Mechanical Properties of Green Concrete Incorporating Waste Foundry Sand Based on Gene Expression Programming.” *Journal of Hazardous Materials* 384: 121322. doi:10.1016/j.jhazmat.2019.121322.

Issa, R. R., and D. Fletcher. 1993. “NEUROFLOR: A Flooring System Selection Neural Network.” *Computing in Civil and Building Engineering; ASCE* 1125–1132. doi:10.13140/R.2.1.1281.6964.

Kang, H. T., and C. J. Yoon. 1994. “Neural Network Approaches to Aid Simple Truss Design Problems.” *Computer-Aided Civil and Infrastructure Engineering* 9 (3): 211–218. doi:10.1111/j.1467-8667.1994.tb00374.x.

Liu, Q. F., M. F. Iqbal, J. Yang, X. Y. Lu, P. Zhang, and M. Rauf. 2021. “Prediction of Chloride Diffusivity in Concrete Using Artificial Neural Network: Modelling and Performance Evaluation.” *Construction and Building Materials* 268: 121082. doi:10.1016/j.conbuildmat.2020.121082.

Martí, P., J. Shiri, M. Duran-Ros, G. Arbat, F. R. De Cartagena, and J. Puig-Bargués. 2013. “Artificial Neural Networks Vs. Gene Expression Programming for Estimating Outlet Dissolved Oxygen in Micro-irrigation Sand Filters Fed with Effluents.” *Computers and Electronics in Agriculture* 99: 176–185. doi:10.1016/j.compag.2013.08.016.

MathWorks. 2020a. *Regression Learner App. MATLAB R2020b*, Version 9.9.0. Natick, MA: MathWorks.

MathWorks. 2020b. *MATLAB R2020b*, Version 9.9.0. Natick, MA: MathWorks.

Messner, J. I., V. E. Sanvido, and S. R. Kumara. 1994. “StructNet: A Neural Network for Structural System Selection.” *Computer-Aided Civil and Infrastructure Engineering* 9 (2): 109–118. doi:10.1111/j.1467-8667.1994.tb00366.x.

Mukherjee, A., and J. M. Deshpande. 1995. “Modeling Initial Design Process Using Artificial Neural Networks.” *Journal of Computing in Civil Engineering* 9 (3): 194–200. doi:10.1061/(ASCE)0887-3801(1995)9:3(194).

Nguyen, D. H., and W. K. Hong. 2019. “Part I: The Analytical Model Predicting Post-Yield Behavior of Concrete-Encased Steel Beams considering Various Confinement Effects by Transverse Reinforcements and Steels.” *Materials* 12 (14): 2302. doi:10.3390/ma12142302.

Shrestha, Y. R., V. Krishna, and G. Von Krogh. 2021. “Augmenting Organizational Decision-making with Deep Learning Algorithms: Principles, Promises, and Challenges.” *Journal of Business Research* 123: 588–603. doi:10.1016/j.jbusres.2020.09.068.

Srinivasulu, S., and A. Jain. 2006. “A Comparative Analysis of Training Methods for Artificial Neural Network Rainfall–runoff Models.” *Applied Soft Computing* 6 (3): 295–306. doi:10.1016/j.asoc.2005.02.002.

Standard, A. A. 2019. *Building Code Requirements for Structural Concrete (ACI 318–19).* In *American Concrete Institute*