Torsional capacity evaluation of RC beams using an improved bird swarm algorithm optimised 2D convolutional neural network

Yang Yu[a,*], Shiwei Liang[b], Bijan Samali[a], Thuc N. Nguyen[c], Chenxi Zhai[d], Jianchun Li[c], Xingyang Xie[b]

[a] *Centre for Infrastructure Engineering, Western Sydney University, Penrith, NSW 2747, Australia*

[b] *Taizhou Customs Comprehensive Technical Service Center, Taizhou, Zhejiang 318000, China*

[c] *School of Civil and Environmental Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia*

[d] *Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, USA*

Abstract

This study presents the application of deep learning technology in torsional capacity evaluation of reinforced concrete (RC) beams. A data-driven model based on 2D convolutional neural network (CNN) is established, where model inputs contain the beam width, beam height, stirrup width, stirrup height, concrete compressive strength, steel ratio of longitudinal reinforcement, yield strength of longitudinal reinforcement, steel ratio of transverse reinforcement, yield strength of transverse reinforcement and stirrup spacing. To enhance the prediction accuracy of the proposed model, an improved bird swarm algorithm (IBSA) is leveraged to optimise the hyperparameters of CNN in the training phase. A comprehensive dataset, comprising 268 groups of laboratory tests of RC beams collected from published articles, is used for model development and validation. The results show that the proposed 2D CNN with hyperparameter optimisation exhibits high performance in predicting torsional strength of RC beams, which outperforms other machine learning models, building codes and empirical formula in terms of a series of evaluation metrics.

Keywords

Torsional strength, reinforced concrete (RC) beams, convolutional neural network, bird swarm algorithm

1. Introduction

Torsion is a common force form that frequently exists in reinforced concrete (RC) structures . In the past, the steel used in RC members was more regular, so the torsional effect is not obvious compared with the bending moment and shear force, which can be tackled by introducing stirrups or increasing the ratio of longitudinal reinforcement. However, with the advance in construction technology, the structural form started to become complex and varied, making the structural members subjected to large torque. On the other hand, people often ignore the torque effect of components in the design, which leads to the serious torsion of structural components, perhaps causing hidden danger or even structural failure.

With regard to this issue, several studies were conducted on torsional behaviour of RC members under pure torsion load or combined torsion and other types of load, which contributed to design codes and prediction formulae/models for evaluating torsional behaviour of RC structures. Hsu carried out a comprehensive experimental study on normal and reinforced concrete members subjected to pure torsion [1]. The experimental results demonstrated that the torsional capacity of RC members is related to several structural parameters, including member dimension, dimension of longitudinal and transverse reinforcements, yield strength of reinforcement and concrete compressive strength. Chiu et al. conducted an experimental investigation on torsional behaviour of normal-strength and high-strength RC beams [2]. Chalioris and Karayannis studied the influences of the ratio of width to height, stirrup number and longitudinal reinforcement location on torsional behaviour of RC beams with rectangular cross-sections [3]. Rahal proposed a simplified non-iterative formulation to predict ultimate torsional resistance of normal-strength and high-strength concrete reinforced beams, which was validated by the test results of 152 rectangular RC beams collected from published articles [4]. The proposed empirical model was also applicable to the RC beam with other shapes of cross-section. Fiore et al. proposed a novel hybridised approach for ultimate torsional strength prediction of RC beams, based on evolutionary polynomial regression [5]. This innovative regression method integrates the characteristics of traditional regression

* Corresponding author
E-mail address: yang.yu@westernsydney.edu.au (Y. Yu)

approaches and gene expression programming, which utilises symbolics to establish the prediction formula. Arslan firstly utilised artificial neural networks (ANNs) to predict torsional strength of RC beams, in which beam dimension, stirrup dimension, spacing, compressive strength of concrete, stirrup yield strength, cross-section of stirrup, steel ratios of both stirrup and longitudinal reinforcement were employed as network inputs [6]. On the basis of Arslan's work, Ilkhani et al. analysed the importance of each input variable on the network output, and then used trained ANN to produce empirical design formula for torsional strength estimation of RC beams [7]. Haroon et al. adopted principal component analysis (PCA) and stacked autoencoder (SAE) to process the training dataset of RC beams to remove the redundant information from input variables, which is able to improve the accuracy of torsional strength prediction [8]. The validation results show that PCA and SAE are effective tools to extract the features from input variables. However, this type of hand-crafted feature perhaps works for a special case or a group of data, but fails to reach a similar performance in other cases.

In the past few years, deep learning technology, with the fusion of feature extraction and pattern classification, has been rapidly developed and applied in various areas, including object recognition [9, 10], natural language processing [11], cancer diagnosis [12, 13], and flow detection [14], etc. Recently, deep learning was also utilised to resolve structural engineering problems, such as surface crack identification [15], structural defect detection [16, 17] and mechanical property prediction [18]. Pathirage et al. employed a deep autoencoder network-based framework for structural damage detection, in which the compressed frequency response functions (FRFs) of vibration measurements by principal component analysis (PCA) were used as model inputs [19]. Fan et al. proposed a novel method based on generative adversarial network (GAN) for dynamic response reconstruction of structures subjected to extreme loads [20]. Jo and Jadidi utilised deep belief networks to develop an automated crack classification system, which can be installed in remotely piloted aircraft (RPA) to analyse RGB and infrared images for structural surface crack detection [21]. Yu et al. put forward an innovative learning approach based on convolutional neural networks with deep architecture for damage diagnosis of structures equipped with smart control devices [22]. Among different deep learning approaches, CNNs were widely utilised in various engineering applications due to outstanding generalisation capability. However, the network performance is related to the setting of hyperparameters, which is a set of parameters to determine the network training. Different assignments of hyperparameters may cause obviously dissimilar performance of CNN. A group of optimal CNN hyperparameters of one application may fail to generate the same performance for others. Accordingly, optimising the CNN hyperparameters is greatly necessary, before the model is employed in specific applications

A new bio-inspired optimisation algorithm, namely bird swarm algorithm (BSA), was recently proposed by Meng et al. [23], which simulates the food-seeking behaviour of birds, such as foraging, vigilance and flying. Due to the advantages of fast convergence and fewer tunable parameters, BSA has been widely utilised in dealing with a variety of engineering optimisation tasks, such as power train optimisation [24], heart disease detection [25], target edge detection [26], wind speed forecast [27], node deployment of wireless sensor networks (WSN) [28], fault diagnosis [29], etc. Malathy et al. employed BSA to deal with optimal node deployment problems in wireless sensor networks (WSN), which aims to reduce energy consumption and prolong the network lifetime [28]. Wu et al. improved BSA via introducing non-dominated sorting method, which was then used for optimising gear ratios of vehicle power train [24]. Niu et al. utilised BSA to optimize penalty factor and kernel parameter of support vector machine (SVM), and this hybrid model was applied to intelligent fault detection of planetary roller screw mechanism [29]. Pruthi et al. combined BSA and fuzzy reasoning to detect object edge from digital images, and test results validated that the proposed approach performs well for identifying smooth, thin and continuous edges, even if the images are polluted with high-intensity noise [26]. BSA was also used to optimise parameters of empirical wavelet transform and least square support vector machine for short-term wind speed prediction [27]. In addition to the wide application, BSA has been proved to be better than particle swarm optimisation (PSO), genetic algorithm (GA), bacterial foraging algorithm (BFO) and ant colony optimisation (ACO) in terms of convergence and solution accuracy, when resolving highly nonlinear optimisation problems [26]. Consequently, BSA can be considered an effective method for CNN hyperparameter optimisation, the research of which has not been reported yet, to the best knowledge of authors.

The objective of this research is to study the potential of using deep learning technology in torsional capacity prediction of RC beams with capability evaluation. A surrogate model based on 2D-CNN is developed accordingly. To guarantee the best prediction performance of the proposed model, BSA is adopted to optimise the network hyperparameters during the training procedure. The essence of hyperparameter optimisation is to resolve a globally minimal optimisation problem, where root mean square error and coefficient of determination between model outputs and corresponding real values of torsional strength are main optimisation targets. Regarding that

standard BSA perhaps falls into local optimum when dealing with highly nonlinear optimisation problems, an improved BSA (IBSA) is put forward, where global location update mechanism and self-adaptive acceleration factors are introduced to promote the global search ability of algorithm. Then, a comprehensive dataset collected from a large number of research articles, containing experimental measurements and tests of RC beams under pure torsional loading, is employed to assess the capacity of trained 2D-CNN via comparing it with other commonly used machine learning methods as well as existing building codes and empirical formula. Eventually, a reliable and robust graphical user interface is designed based on Matlab App Designer to facilitate structural designers to fully utilise this innovative technology for designing RC beams in practice.

2. Theoretic background

2.1. Convolutional neural network

Convolutional neural network (CNN) is a type of feed-forward neural network with deep learning capacity, the structure of which is similar to that of multi-layer perception (MLP). Different from MLP with the challenges of a large number of training parameters and low prediction speed, CNN conducts the improvement in the following aspects: 1) the spare connection is adopted between two adjacent layers, 2) the weights are shared between the neurons in the same layer, and 3) down-sampling is conducted to reduce the feature dimension in the data space. Due to these advantages, CNN can not only keep the high original features that are then transformed into abstract features, but also reduce the network complexity and improve the prediction efficiency. Generally, CNN is made up of input layer, convolution layers, pooling layers, fully connected (FC) layers and output layer, where the convolution and pooling layers are set alternatively, followed by FC and output layers. The detailed information about CNN is presented in the following sub-sections.

2.1.1. Convolution layer

The convolution layer is a key element in CNN, which has excellent feature learning ability. Each convolution layer contains a number of convolution kernels used for feature extraction. Generally, the first convolution layer only extracts lower-level features from original data, while the deeper convolution layers can iteratively extract more complex features from lower-level features, significantly simplifying the feature extraction procedure. In addition, the kernel of convolution layer is only connected with some local areas in the input, and the kernels in the same convolution layer will share the weights and bias, which not only effectively reduces the number of network parameters, but also avoids the overfitting problem. Multiple kernels in each convolution layer can learn different characteristics of data and gain various output data to guarantee the full use of information.

2.1.2. Pooling layer

The pooling layer, also known as down-sampling layers, is used to conduct the sparse processing on the feature map to reduce the amount of data calculation. The outputs of convolution layer are pooled to further reduce the number of model parameters as well as the complexity of network to avoid over-fitting. In the pooling layer, a function is used to replace the actual output of a feature location with overall statistical characteristics of outputs of adjacent features at that location. Commonly used pooling functions include maximum pooling and averaged pooling, where maximum pooling and averaged pooling replace the current value with the maximum and averaged values of all the values in the adjacent area, respectively. Also, the L2 normalization and weighted average method are employed to conduct the pooling operation. The pooling operation can guarantee that the location translation, rotation and scaling of previous layer do not affect the output results. When the inputs satisfy a priori assumption of local invariance, pooling is able to remarkably improve the calculation efficiency of the network. The reduction of input size also helps decrease the number of model parameters, which further saves the computational cost of CNN.

2.1.3. Fully connected layers

The fully connected (FC) layer refers to that each neuron in the previous layer is connected with all the neurons in the following layer. The function of FC is to combine all the local features extracted from convolution and pooling layers into a complete feature graph. In this process, different weights need to be given to different features, and a weight matrix is formed, which is to be tuned during the network training.

2.1.4. Activation function

The activation function is usually placed after convolution and FC layers, which plays a significant role in achieving nonlinear mapping of input features. The most common activation functions involve binary step function, hyperbolic tangent (tanh) function, logistic function, rectified linear unit (ReLU) function, sigmoid shrinkage (Swish) function, Gaussian function, etc. The corresponding mathematical expressions are shown in Eq. (1) to Eq. (6). Fig. 1 compares the curves of different activation functions.

$$binary\ step \qquad Atf(t) = \begin{cases} 0, t < 0 \\ 1, t \geq 0 \end{cases} \tag{1}$$

$$tanh \qquad Atf(t) = \frac{e^{t} - e^{-t}}{e^{t} + e^{-t}} \tag{2}$$

$$logistic \qquad Atf(t) = \frac{1}{1 + e^{-t}} \tag{3}$$

$$ReLU \qquad Atf(t) = \max\{0, t\} \tag{4}$$

$$Swish \qquad Atf(t) = \frac{t}{1 + e^{-t}} \tag{5}$$

$$Gaussian \qquad Atf(t) = e^{-t^{2}} \tag{6}$$

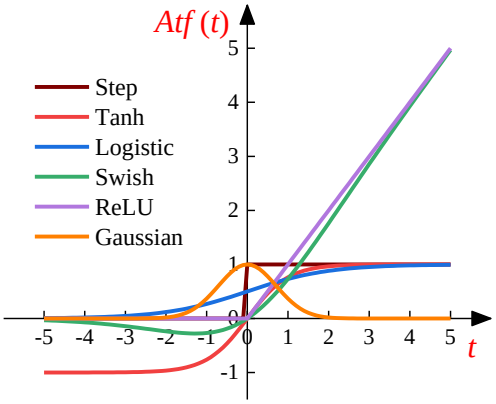


Fig. 1. Activation functions of CNN

2.2. Bird swarm algorithm

Bird swarm algorithm (BSA) is a novel swarm-based heuristic intelligent optimisation algorithm, which was designed based on the foraging, vigilance and flight behaviors of birds. Compared with traditional swarm optimisation algorithms, BSA has the benefits of quick convergence, fewer tunable parameters and good robustness. The studies in [26] have sufficiently demonstrated that BAS outperforms particle swarm optimisation (PSO) and differential evolution in the optimisation of engineering application problems. Suppose the bird swam is composed of N_b birds flying in an M-dimensional space, and their behaviors can be simplified as the following idealised rules.

Rule 1. Each bird in the swarm is able to randomly switch between two behaviors of foraging and vigilance. For an arbitrary bird, if the random number *rn* is more than the probability of vigilance, the bird chooses the foraging behaviour. Or else, it selects the behaviour of vigilance.

Rule 2. During the foraging process, each bird timely records and updates itself best location and global best location of swarm for food searching, which can be formulated in Eq. (7).

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \left(gb_j - x_{i,j}^{t}\right) \cdot SA \cdot rd_1 + \left(pb_{i,j} - x_{i,j}^{t}\right) \cdot CA \cdot rd_2 \tag{7}$$

where t denotes current iteration number; i corresponds to the bird index and $i = 1,\ldots, N_b$; j corresponds to the solution dimension and $j = 1,\ldots, M$; gb denotes the global best location of swarm; SA denotes the social acceleration factor; pb_i denotes individual best location of ith bird; CA denotes the cognitive acceleration factor; and rd_1 and rd_2 denotes two random numbers between 0 and 1.

Rule 3. During the vigilance process, each bird attempts to move towards the swarm centre. The birds with lower fitness values (more food reserve) are closer to the swarm centre than the birds with higher fitness values. This behavior can be formulated as follows.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \left(ave_j - x_{i,j}^{t}\right) \cdot B_1 \cdot rd_{[0,1]} + \left(pb_{k,j} - x_{i,j}^{t}\right) \cdot B_2 \cdot rd_{[-1,1]} \tag{8}$$

$$B_1 = \beta_1 \cdot e^{-\frac{N_b \cdot pb_i}{\sum_{i=1}^{N_b} pb_i + \tau}} \tag{9}$$

$$B_2 = \beta_2 \cdot e^{\frac{pb_i - pb_k}{|pb_i - pb_k| + \tau} \cdot \frac{N_b \cdot pb_i}{\sum_{i=1}^{N_b} pb_i + \tau}} \tag{10}$$

where k is a randomly selected integer in the range of [1, N_b], and $k \neq i$; β_1 and β_2 represent two constant numbers in the range of [0, 2]; ave_j denotes averaged value of optimal location of jth dimension of swarm; pb_i denotes the best fitness of ith bird; τ is a small number to prevent zero-division; $rd_{[0,1]}$ and $rd_{[-1,1]}$ are two random numbers in the ranges of [0, 1] and [-1, 1], respectively.

Rule 4. The birds fly from one place to another with a regular frequency RF. In the meantime, the roles of birds switch between producer and scrounger based on their food reserve. The birds with lower food reserves are deemed as the scroungers, while the other birds are regarded as the producers. The producers are responsible for foraging, while the scroungers follow the producers to find the food. The food searching behaviors of producers and scroungers could be formulated in Eq. (11) and Eq. (12), respectively.

$$x_{i,j}^{t+1} = (1 + \alpha_1) x_{i,j}^{t} \tag{11}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \alpha_2 \left(x_{k,j}^{t} - x_{i,j}^{t}\right) fp \tag{12}$$

where α_1 is a random number with Gaussian distribution; α_2 is a random number with uniform distribution between 0 and 1; fp is a parameter to indicate the degree that the scrounger follows the producer for finding food.

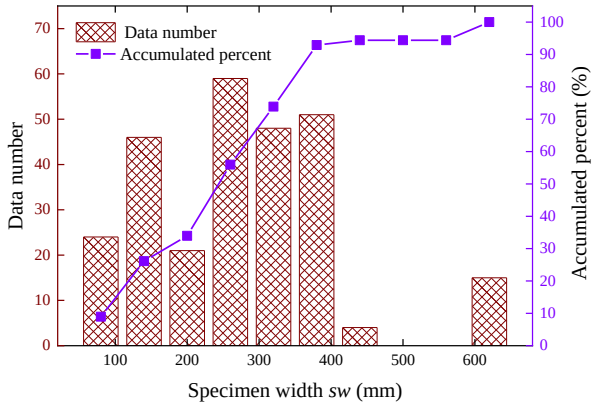
3. Experimental data collection and analysis

To develop a highly accurate data-driven model to predict torsional capacity of RC beams, it is crucial to establish a comprehensive dataset that contains all the potential influence factors. In this research, 268 groups of experimental measurement and test results of RC beams with rectangular cross-sections were collected from a large number of articles published between 1985 and 2019, which make up the dataset used for model development and validation [1, 30-44]. The dataset contains the torsional strength and its influencing factors, i.e. the dimensions and mechanical properties of concrete beam and reinforcements. Here, ten structural parameters are selected as the inputs of the model to be developed, i.e., the width of beam specimen (sw), height of beam specimen (sh), width of stirrup (stw), height of stirrup (sth), compressive strength of concrete (sc), steel ratio of longitudinal reinforcement (lrr), yield strength of longitudinal reinforcement (syl), steel ratio of transverse reinforcement (tra), yield strength of transverse reinforcement (syt) and stirrup spacing (sts), which were obtained from the specimens of RC beam under pure tension tests. The statistical information of these parameters is summarised in Table 1, including the values of maximum, minimum, mean, median, standard deviation, kurtosis and skewness. It is clearly seen that each parameter has a relatively broad range, which is beneficial to the robustness of developed surrogate model. Fig. 2 presents the column diagram for statistical distribution of each parameter from the dataset.

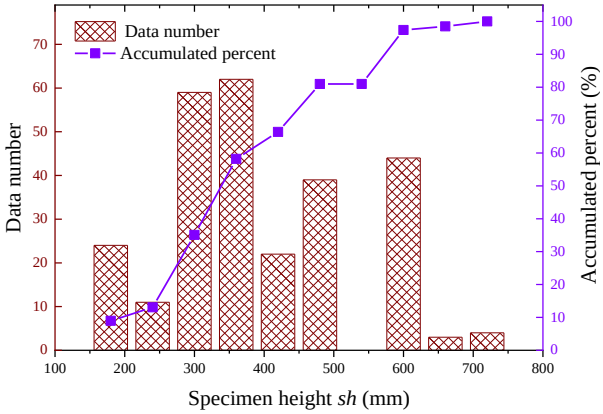
Table 1 Statistical information of each parameter of RC beam from the dataset

Variable	Statistical index						
	Maximum	Minimum	Median	Mean	Standard deviation	Kurtosis	Skewness
sw	600.00	85.00	254.00	273.23	120.53	3.84	0.83

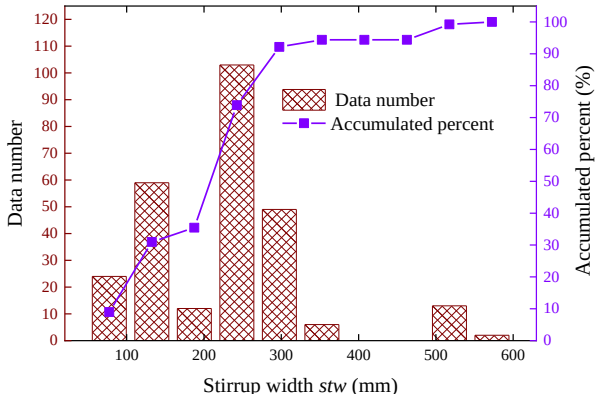
sh	700.00	178.00	381.00	399.54	133.45	2.11	0.34
stw	546.00	56.50	216.00	223.53	108.45	4.50	1.00
sth	650.00	149.50	327.00	342.68	121.99	2.18	0.44
sc	109.80	14.30	40.25	47.32	21.24	2.89	0.94
lrr	3.89	0.18	1.00	1.33	0.95	3.36	1.25
syl	724.00	309.00	400.00	444.42	121.31	2.36	0.77
tra	3.20	0.13	0.91	0.99	0.53	4.16	1.01
syt	715.00	265.00	385.00	437.87	128.05	2.29	0.81
sts	300.00	41.00	92.00	106.01	43.15	6.05	1.65
ts	239.00	2.18	39.40	54.32	44.85	3.21	0.92



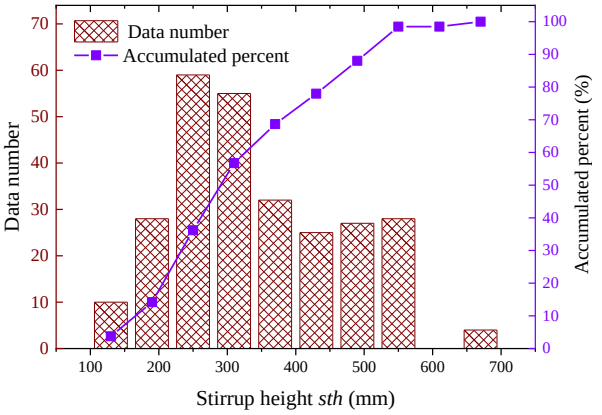
(a) Width of beam specimen



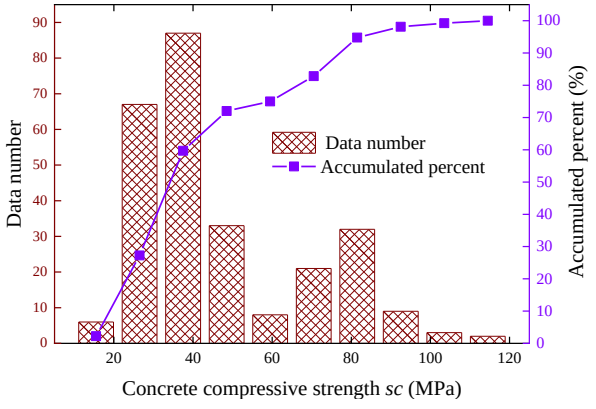
(b) Height of beam specimen



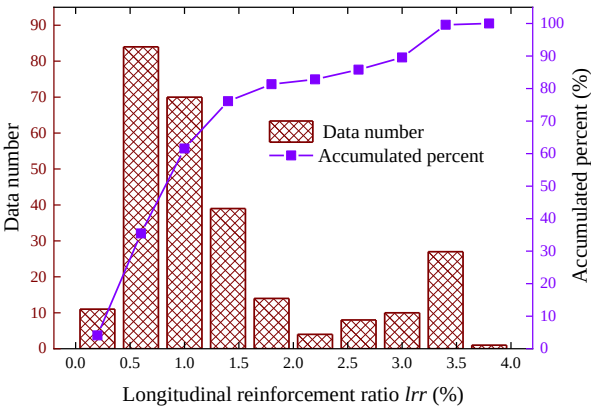
(c) Width of stirrup



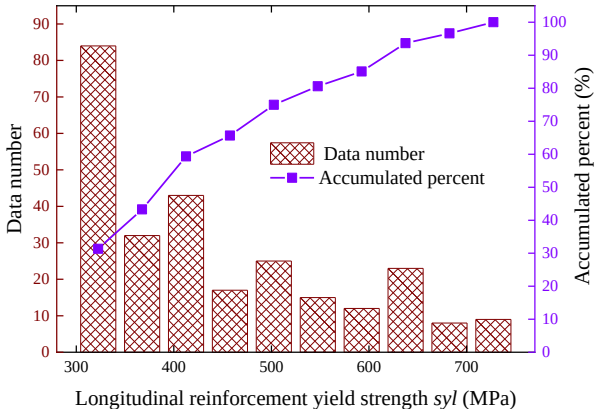
(d) Height of stirrup



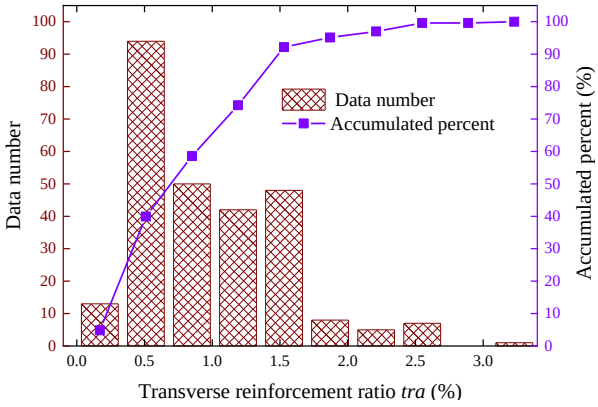
(e) Concrete compressive strength



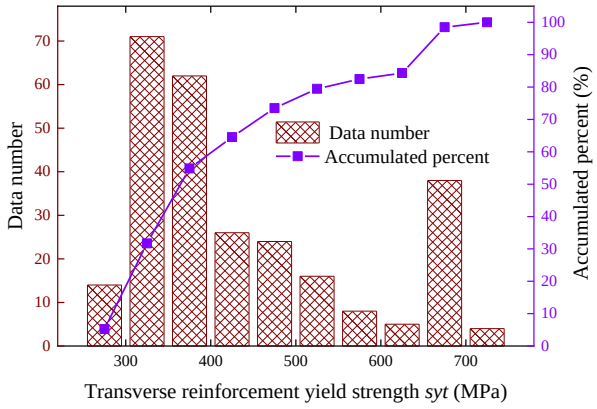
(f) Steel ratio of longitudinal reinforcement



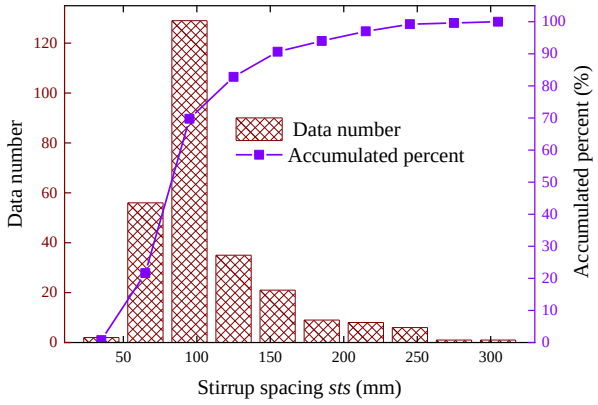
(g) Yield strength of longitudinal reinforcement



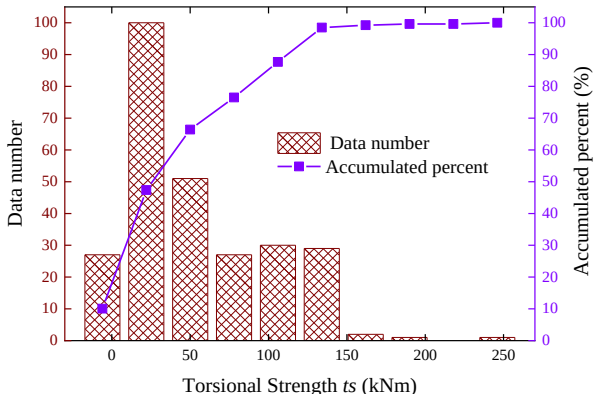
(h) Steel ratio of transverse reinforcement



(i) Yield strength of transverse reinforcement



(j) Stirrup spacing



(k) Torsional strength

Fig. 2. Statistical distributions of input and output parameters

From the view of machine learning study, each model input should be independent of other inputs to avoid information redundancy, which can degrade the performance of the trained model. Hence, it is necessary to conduct a sensitivity analysis on all the influencing parameters of the torsional strength of RC beams. Here, the correlation coefficient between arbitrary two inputs is employed as the metric to evaluate their dependency degree. In general, the correlation coefficient between any two input variables is in the range of [-1, 1]. More than 0.8 or less than -0.8 of correlation coefficient indicates high correlation between two inputs. In that case, one input is suggested to be deleted, since it can be expressed by linear combinations of other inputs. In this work, a correlation coefficient matrix with the size of 10 × 10 is established to indicate the dependency of 10 input variables related to torsional strength of RC beams, the result of which is shown in Fig. 3. It is clearly seen that except the values

of elements on the diagonal of the matrix that represent the self-correlation coefficients, all the values of elements in the matrix are between -0.7 and 0.7, demonstrating low dependency of input variables in the dataset of RC beams. Accordingly, these ten parameters of RC beams can be utilised as the inputs to develop a data-driven model for torsional capacity prediction.

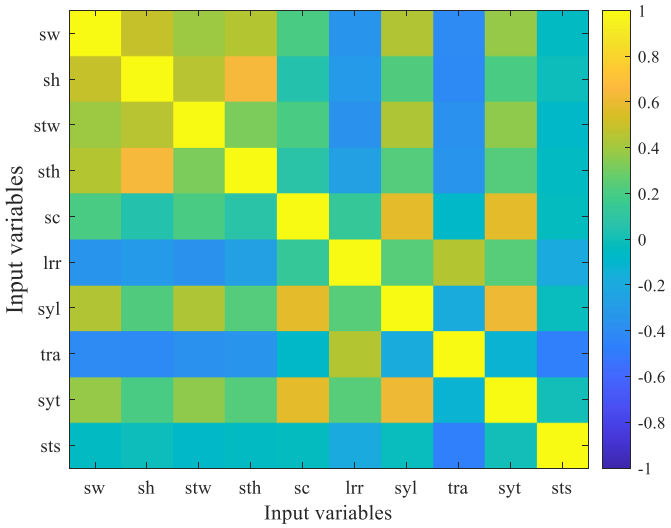


Fig. 3. Dependency analysis of ten input variables.

4. Methodology

4.1. Framework of proposed model for torsional capacity prediction of RC beams

In this research, a novel data-driven model based on 2D-CNN is developed to predict the torsional capacity of RC beams, the architecture of which is shown in Fig. 4. It consists of a 2D input layer, two convolution layers, a pooling layer, two FC layers and an output layer. The size of input layer is 3×4, containing 10 variables of sw, sh, stw, sth, sc, lrr, syl, tra, syt and sts, and 2 padded zeros. The first convolution layer contains 32 kernels with the kernel size of 2×2, stride size of 1 and padding size of 1. Following the first convolution layer, a maximum pooling layer is added to down-sample the learned representation, which has been proved to be better than average pooling [45]. The kernel size, stride size and padding size of pooling layer are 2×2, 1 and 0, respectively. Next, a convolution layer, which contains 64 kernels with the kernel size of 2×2, stride size of 1 and padding size of 1, is added to explore the deep features that are sensitive to torsional strength of RC beams. Then, two FC layers are connected to Con2, denoted by Fc1 and Fc2. Fc1 is utilised to flatten 2D representations by a 1D feature map, the outputs of which are regarded as inputs of Fc2. Fc2 is the second FC layer that is designed to compact flattened features and reduce the dimension. Because the neuron numbers of Fc1 and Fc2 are 30 and 10, respectively, which is apparently in a reduction manner, they are deemed as hidden layers for optimal feature selection during the process of mapping. Accordingly, redundant and useless features are removed, when the outputs of Fc2 are employed as inputs for eventual torsional capacity evaluation. In addition, to avoid the trained network being overfitted owing to multiple hidden neurons in Fc1 and Fc2, the dropout operation is conducted between Con2 and Fc1, which can increase the diversity of CNN via randomly restraining some neurons. The detailed information of each layer in proposed 2D-CNN has been concluded in Table 2.

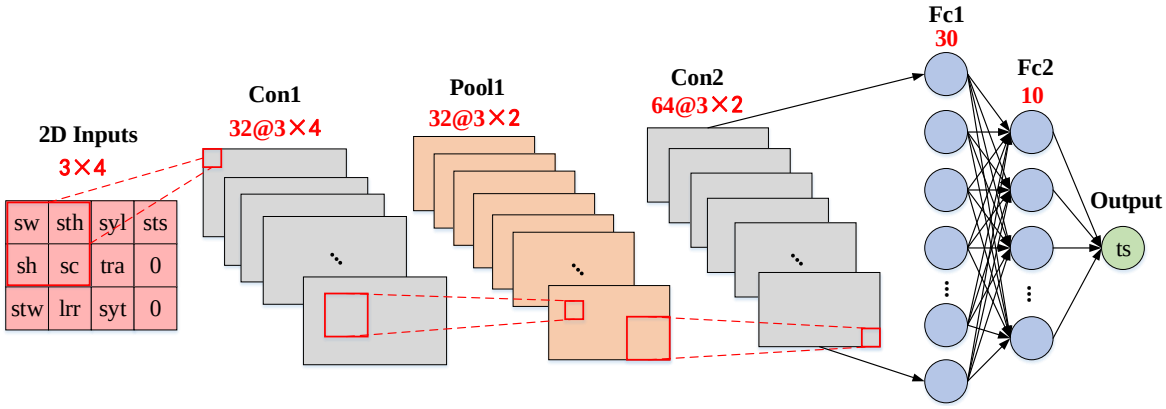


Fig. 4. Architecture of proposed 2D-CNN for torsional strength prediction of RC beams.

Table 2 Parameter information of each layer in 2D-CNN

Layer name	Kernel size	Padding size	Stride size	Output dimension
Con1	2×2	1	1	32
Pool1	2×2	0	1	32
Con2	2×2	1	1	64
Fc1	—	—	—	30
Fc2	—	—	—	10
Regression	—	—	—	1

4.2. 2D-CNN model training

The training of proposed 2D-CNN is based on supervised learning, where a loss function is needed to measure the difference between real value and prediction value from network. For the model training, the main target is to minimize the value of loss function via tuning the network parameters. In this study, the 2D-CNN is developed to predict the torsional capacity of RC beams, which is a regression problem in essence. Accordingly, we employ the mean square error (MSE) between real and predicted torsional strengths as the loss function, which can be regarded as a calculation method Euclidean distance, defined as follows.

$$L(Y,\hat{Y}) = \frac{1}{N_{tr}}\sum_{k=1}^{N_{tr}}(\hat{y}_i - y_i)^2 \tag{13}$$

where N_{tr} denotes the total number of training data, and $\hat{y}_i$ and y_i denote the predicted and real torsional strengths, respectively.

Then, the stochastic gradient descent with momentum (SGDM) method is used to minimize the loss value via a procedure of error back-propagation. Before the network arrives at convergence, the weights and bias of each layer are continuously tuned by SGDM to diminish the network error. The parameter update formula is displayed in Eq. (14).

$$MW_{t+1} = MW_t - ltv_t \cdot \Delta L(MW_t; Y, \hat{Y}) \tag{14}$$

where MW denotes the parameter vector consisting of weights and bias, ltv denotes the learning rate, and ΔL denotes the gradient of loss function.

4.3. Hyperparameter optimisation of proposed model using improved BSA

As all we know, the performance of 2D-CNN is heavily dependent on the assignment of network hyperparameters. Improper setting of hyperparameters may cause the trained network to be either overfitted or underfitted. The same network can generate entirely distinct capacities using different values of CNN hyperparameters. Hence, it is necessary to set optimal values of hyperparameters to achieve the best model for torsional capacity prediction of RC beams. Here, the hyperparameters of 2D-CNN model include initial learning rate (α_l), momentum (γ), L2-regularization factor (l_2-R), learning rate drop period (T_{LRD}), learning rate drop factor

(ξ_{LRD}) and dropout factor (χ). The ranges and types of these hyperparameters are summarised in Table 3. Optimising hyperparameters of 2D-CNN can be considered as resolving a minimisation problem, where the optimisation targets are error and similarity degree between real torsional capacity and prediction from model. In this study, the fitness function is composed of two components, root mean squared error (RMSE) and coefficient of determination (R^2) between real and predicted values. The definition of fitness function and relevant mathematical expressions are displayed as follows.

$$FitF = \frac{Fit_{RMSE}}{Fit_{R^2}} \tag{15}$$

$$Fit_{RMSE} = \sqrt{\frac{1}{N_{td}} \sum_{k=1}^{N_{td}} [ts_p(k) - ts_o(k)]^2} \tag{16}$$

$$Fit_{R^2} = \left[\frac{N_{td}\left(\sum_{k=1}^{N_{td}} ts_p(k) ts_o(k)\right) - \left(\sum_{k=1}^{N_{td}} ts_o(k)\right)\left(\sum_{k=1}^{N_{td}} ts_p(k)\right)}{\sqrt{\left[N_{td} \sum_{k=1}^{N_{td}} ts_p(k)^2 - \left(\sum_{k=1}^{N_{td}} ts_p(k)\right)^2\right]\left[N_{td} \sum_{k=1}^{N_{td}} ts_o(k)^2 - \left(\sum_{k=1}^{N_{td}} ts_o(k)\right)^2\right]}} \right]^2 \tag{17}$$

where N_{td} denotes the number of training data, and ts_o and ts_p represent real (observed) and predicted torsional strengths of RC beam, respectively. The lower the fitness value, the better the solution of hyperparameters. If the fitness is close to 0, the corresponding solution can be regarded as the optimal solution of hyperparameters of 2D-CNN for predicting torsional capacity. To deal with this optimisation task, the BSA is employed to tune the hyperparameters during the process of model training.

Table 3 Boundaries and types of hyperparameters

Parameter	Upper limit	Lower limit	Type
α_l	3e-4	1e-5	Decimal
γ	0.98	0.8	Decimal
l_2-R	1e-2	1e-10	Decimal
T_{LRD}	30	10	Integral
ξ_{LRD}	0.3	0.05	Decimal
χ	0.15	0.8	Decimal

Even though the BSA has been proved to be effective in function optimisation, it still suffers from the problems of low solution accuracy and premature when tackling complex optimisation problems with high nonlinearity like Eq. (15) in this research. The main reason leading to low accuracy and premature is randomness and fixed length in the location update formulae in the standard BSA. For instance, in the behavior of vigilance, *k*th bird is randomly selected for the optimal location update. If the current *i*th individual has better fitness and the selected *k*th bird has the worse fitness, the algorithm will generate numerical oscillation during the evolutionary iteration, thus increasing the calculation time and reducing the search ability of bird. In addition, in the behavior of foraging, the social and cognitive acceleration factors are always constants for updating the location of birds. The lower values of both acceleration factors can result in slow convergence in the early iteration of algorithm, while the higher values may cause the algorithm to fall into the local optimum.

To resolve these challenges in standard BSA, this study puts forward an improved BSA (IBSA) to optimise the hyperparameters of 2D-CNN for accurately predicting torsional capacity of RC beams, where two modifications are applied in the location update formulae. The first modification is to replace *k*th randomly selected bird with global optimal individual in the swarm, which effectively enhances the targeting of optimisation. Hence, Eq. (8) and Eq. (10) are reformulated as follows.

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \left(ave_j - x_{i,j}^{t}\right) \cdot B_1 \cdot rd_{[0,1]} + \left(gb_j - x_{i,j}^{t}\right) \cdot B_2 \cdot rd_{[-1,1]} \tag{18}$$

$$B_2 = \beta_2 \cdot e^{\frac{pb_{i,j} - gb_j}{|pb_{i,j} - gb_j| + \tau} \cdot \frac{N_b \cdot pb_{i,j}}{\sum_{i=1}^{N_b} pb_{i,j} + \tau}} \tag{19}$$

where gb_j denotes the globally optimal location of jth dimension. The second modification is the introduction of self-adaptive acceleration factors, which can adjust the values at each iteration. Essentially, the larger acceleration factors are required in the early stage of iteration to enhance the exploration ability of algorithm, while the smaller acceleration factors are preferred in the later stage of evolution to improve the exploitation ability of algorithm for solution refinement. Therefore, the decreasing acceleration factors are proposed in IBSA, the definitions of which are shown in Eq. (20) and Eq. (21)

$$SA(i_c) = SA_{min} + (SA_{max} - SA_{min})e^{-\mu_1\left(\frac{i_c}{TN}\right)^{\delta_1}} \tag{20}$$

$$CA(i_c) = CA_{min} + (CA_{max} - CA_{min})e^{-\mu_2\left(\frac{i_c}{TN}\right)^{\delta_2}} \tag{21}$$

where i_c denotes the current iteration number; SA_{min} and SA_{max} are minimum and maximum social acceleration factors, respectively; CA_{min} and CA_{max} are minimum and maximum cognitive acceleration factors, respectively; TN denotes maximum iteration number of algorithm. Fig. 5 presents the curve of proposed nonlinear acceleration factor via a comparison with linearly decreasing curve. It is clearly noted that the curves of nonlinear factors are capable of keeping higher value for a longer time and quickly decline to lower value, compared with linearly decreased factor. Among three nonlinear curves in the figure, the green one with μ=10 and δ=8 is the option in Eq. (20) and Eq. (21) because of symmetry of curve that balances the global and local search capabilities of algorithm.

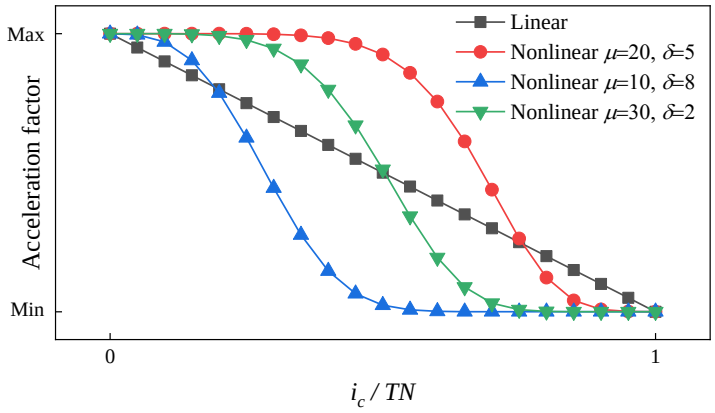


Fig. 5. Adaptively decreased acceleration factor.

Based on proposed IBSA, the process of hyperparameter optimisation of 2D-CNN can be summarised in the following steps.

Step 1. Confirm the dimension of solution, and set algorithm parameters of IBSA, such as swarm population, maximum iteration number, minimum and maximum social acceleration factors, minimum and maximum cognitive acceleration factors, β_1 and β_2, etc.

Step 2. Initialise the bird swarm and set the current iteration number t is 1. Then, calculate the fitness of each bird in the swam and save individual optimum and global optimum.

Step 3. Check the birds' conditions. According to different conditions of birds, use corresponding equations to update the birds' locations.

Step 4. Calculate the fitness of each bird. Then, compare the current individual and global optimums with previous ones. If the current values are better than previous ones, update $pb_{i,j}$ and $gb_{,j}$.

Step 5. Check the termination condition. If the current iteration number t is more than maximum iteration number, terminate the algorithm evolution. Or else, $t = t + 1$ and return to Step 3.

A detailed flowchart for hyperparameter optimisation of proposed 2D-CNN for torsional capacity evaluation of RC beams using IBSA is illustrated in Fig. 6.

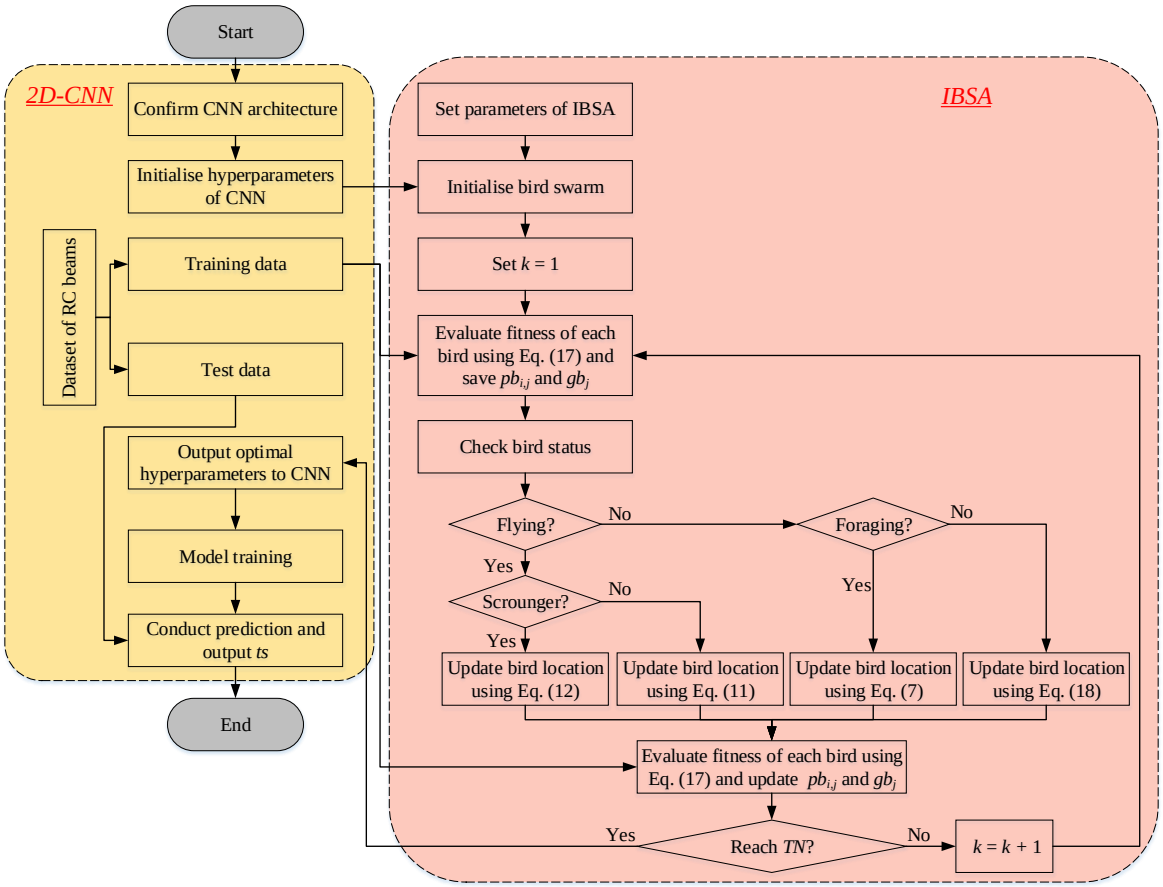


Fig. 6. Flowchart of hyperparameter optimisation of 2D-CNN using IBSA.

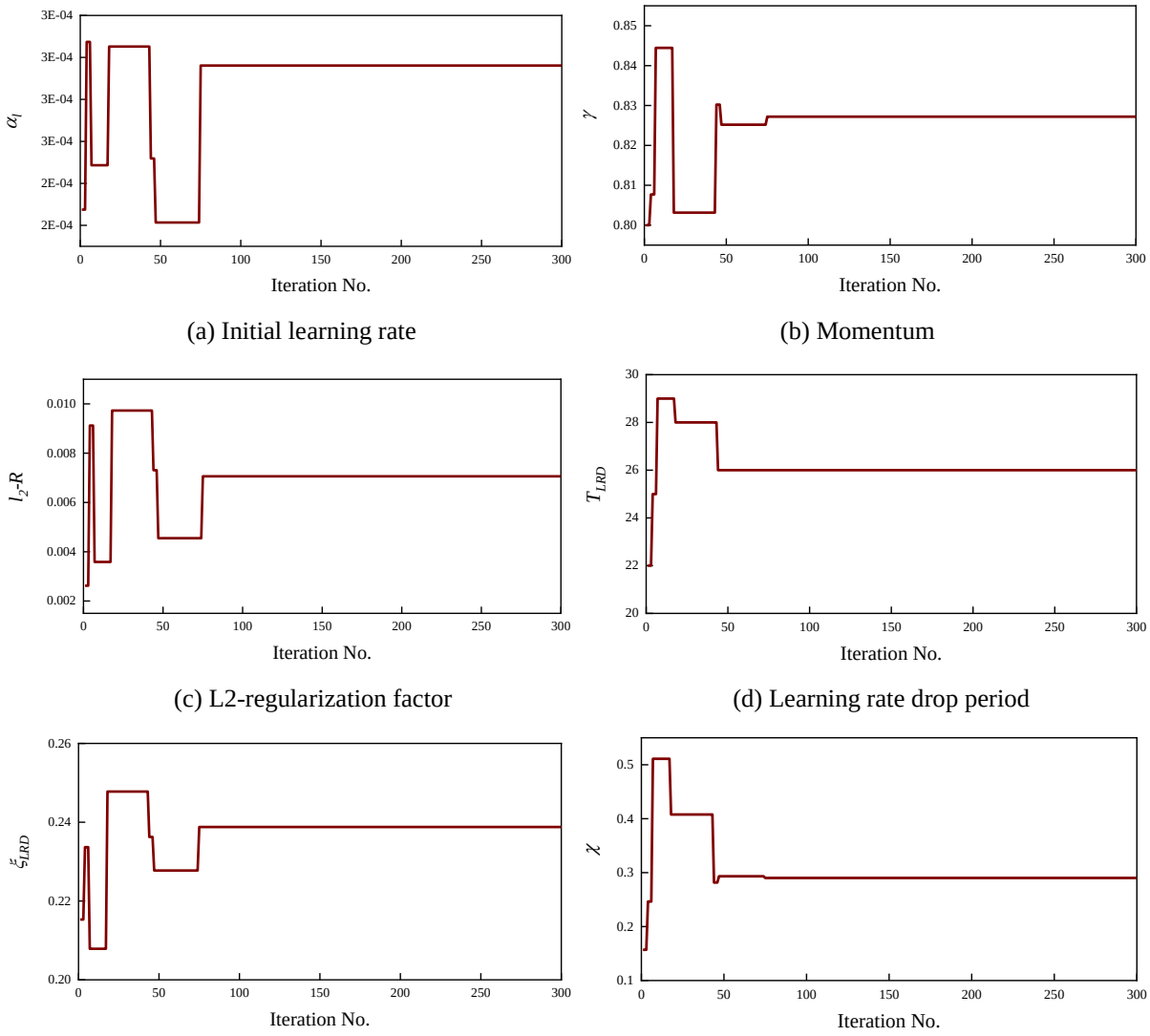
5. Results and discussion

Based on proposed IBSA optimised 2D-CNN (IBSA-2D-CNN), the predictive model for torsional strength of RC beams is established based on collected dataset. Here, the whole dataset is randomly segmented into two groups, where the first group contains 188 samples (70%) that are used to train the proposed model and the rest 80 samples (30%) are used as test data to validate the performance of the trained model. The coding of both IBSA and 2D-CNN is based on Matlab v.2021a, where the Deep Learning Toolbox is employed. The program is implemented on a laptop with Intel i5-8350U CPU, 8GB memory (RAM) and 64-bit operating system. GPU method is applied to the training procedure to accelerate the optimisation of hyperparameters of 2D-CNN using IBSA. In this study, the parameters of IBSA are set as follows: the swarm population is 40, maximum iteration number is 300, minimum values of social and cognitive acceleration factors are 1, maximum values of social and cognitive acceleration factors are 2, $\beta_1 = 1, \beta_2 = 1, fp \in [0, 2]$, and $RF = 10$.

Fig. 7 portrays the optimisation procedure of each hyperparameter of 2D-CNN using IBSA. It is noted all the hyperparameters reach their optimum in the meantime at around 75 iterations, which sufficiently demonstrates the fast convergence ability of the proposed improved algorithm. The optimisation solutions to the hyperparameters of 2D-CNN for torsional strength prediction of RC beams are displayed in Table 4.

Table 4 Optimisation results of hyperparameters of 2D-CNN

Hyperparameter	α_l	γ	l_2-R	T_{LRD}	ξ_{LRD}	χ
Value	2.68e-4	0.8272	0.0071	26	0.2388	0.2901



(a) Initial learning rate

(b) Momentum

(c) L2-regularization factor

(d) Learning rate drop period

(e) Learning rate drop factor

(f) Dropout factor

Fig. 7. Hyperparameter optimisation results

Furthermore, to elaborate the superiority of proposed optimisation algorithm, a comparative study is conducted by comparing the fitness of IBSA with the fitness values of three commonly used algorithms, i.e., genetic algorithm (GA), particle swarm optimisation (PSO) and BSA, for hyperprameter optimisation of 2D-CNN based on training dataset of RC beams. To carry out an impartial investigation, the population size and maximum iteration numbers of GA, PSO and BSA are set the same as that of IBSA, i.e., 40 and 300, respectively. Other parameters of three optimisation algorithms are set as follows: for GA, the selection method is roulette wheel, mutation rate is 0.015 and the rate of single-point crossover is 0.085; for PSO, the inertia weight is 0.75 and learing rate is 2; for BSA, the social and cognitive acceleration factors are set as 1.5, and other parameters are the same as that of IBSA. The comparison result of four algorithms in terms of fitness change is shown in Fig. 8. It is apparent that GA has the worst performance on 2D-CNN hyperparameter optimisation for predicting torsional strength of RC beams, with the highest fitness value. Although the BSA has the fastest convergence speed among four algorithms, it is premature due to higher fitness than that of PSO and IBSA. The proposed IBSA, however, outperforms PSO in both convergence and solution accuracy, owing to outstanding algorithm exploitation ability. This promising result further proves the effectiveness of the proposed IBSA in hyperparameter optimisation of 2D-CNN.

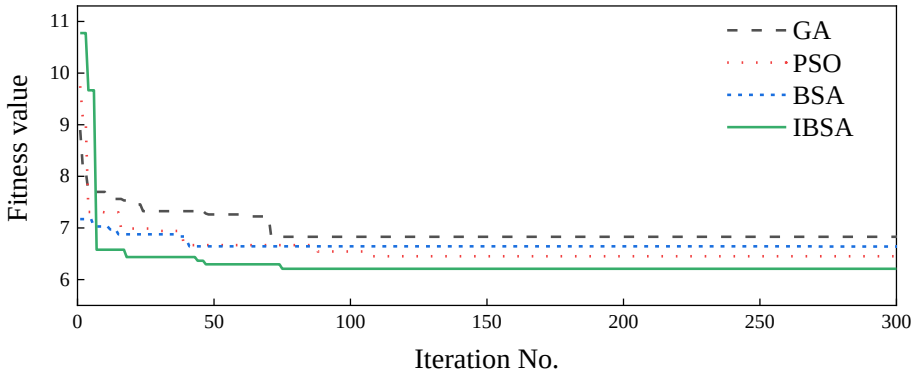
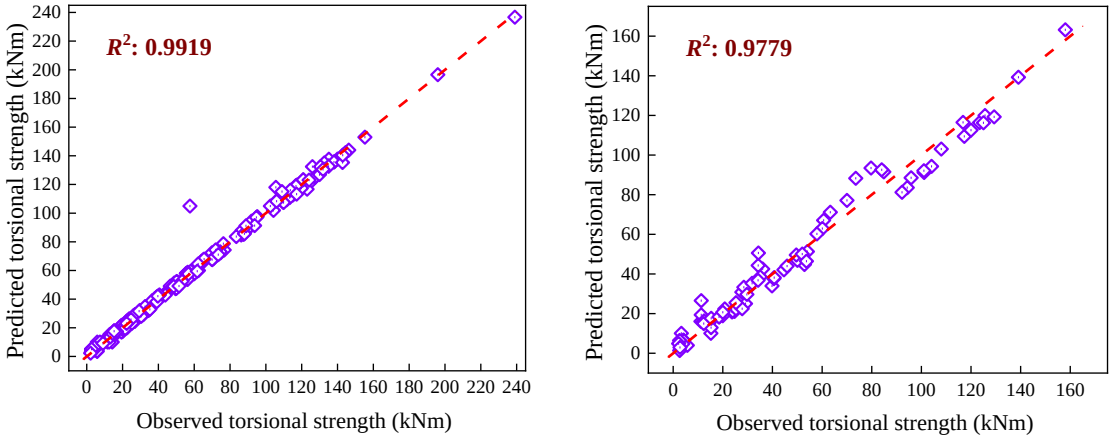


Fig. 8. Fitness comparison of different algorithms

Based on the optimal hyperparameter setting, the surrogate model based on 2D-CNN is constructed and trained to predict the torsional capacity of RC beams. Then, the test data are sent to the trained model for model evaluation via comparing the model predictions with real experimental measurements. Fig. 9 describes the comparison between predicted and experimental torsional strengths for both training and test data. In Fig. 9 (a), it is clearly observed that almost all the points are located on the fitting line of $y = x$, indicating outstanding prediction performance of training data with the R-squared value of 0.9919. From the results in Fig. 9 (b), all the points are scattered around the fitting line of $y = x$, with the R-squared value of 0.9779.



(a) Training (b) Test

Fig. 9. Comparison between real torsional strength and predictions of proposed IBSA-2D-CNN.

To further demonstrate the superiority of proposed IBSA-2D-CNN, a comparative investigation is undertaken by comparing it with other commonly used machine learning approaches as well as existing building codes and empirical formula. The machine learning models used in this comparison contain Gaussian process regression (GPR), support vector machine (SVM), bagged tree (BT), and standard CNN without hyperparameter optimisation. The inputs and output of four machine learning models are the same as that of proposed model. Other information of these models are summarised as follows. For the GRP model, the kernel function is Matern 5/2, and three hyperparameters of feature length scale, data standard deviation and shape factor are set as 100, 33 and 0.0001, respectively. For the SVM model, the radial basis function is employed as kernel function, and three hyperparameters of penalty parameter, kernel parameter and loss factor are set as 15.3742, 0.5685 and 0.0178, respectively. For the BT model, the minimal leaf size is set as 8, and number of learning cycles is set as 30. These parameter values are obtained via the optimisation procedure. For standard CNN, the hyperparameters of initial learning rate, momentum, L2-regularization factor, learning rate drop period, learning rate drop factor and dropout factor are set as 0.01, 0.9, 0.001, 10, 0.1 and 0.5, respectively. The building codes and empirical model include ACI-318-2005 [46], BS-8100 [47] and analytical formulation designed by Rahal [4]. The relevant mathematical expressions are provided as follows.

ACI-318-2005

$$T_n = \frac{2A_o A_t f_{yt}}{s}\cot\theta \tag{22}$$

$$\cot\theta = \sqrt{\frac{A_l f_{yl} s}{A_t f_{yt} p_h}} \tag{23}$$

BS-8110

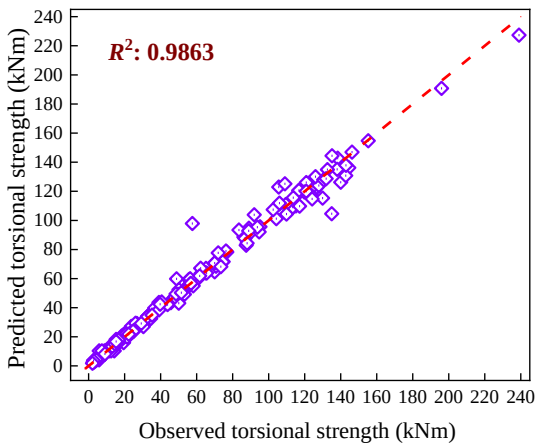
$$T_n = \frac{0.8x_1 y_1 (0.87 f_{yt}) A_{sv}}{s} \tag{24}$$

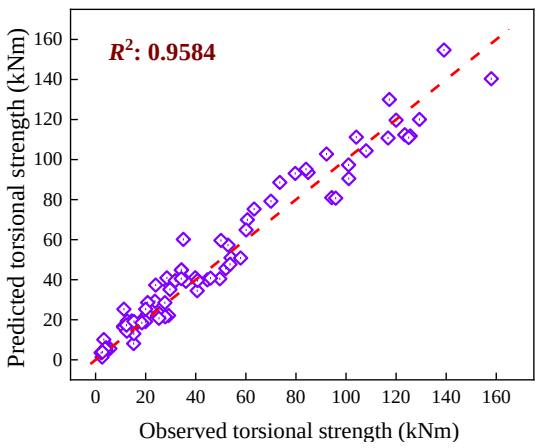
Rahal formula

$$T_n = 0.33(f_c')^{0.16} A_c \left(A_l f_{yl} \frac{A_t f_{yt}}{s}\right)^{0.35} \tag{25}$$

where A_o denotes the cross area enclosed by the shear flow path, which can be taken as $0.85A_{oh}$; A_{oh} denotes the area enclosed within the centreline of the hoops; A_t denotes the cross-sectional area of one-leg of closed stirrup; f_{yt} denotes the yield strength of closed stirrups; A_{sl} denotes the total area of symmetrically distributed longitudinal reinforcement; f_{yl} denotes the yield strength of longitudinal reinforcement; s is the spacing of closed stirrups; θ denotes the angle of compression diagonals; p_h denotes the perimeter of the centreline of the outmost closed transverse torsional reinforcement. x_1 and y_1 denote the centre-to-centre of the shorter and longer legs of closed stirrups; A_{sv} denotes the area of the two legs of stirrups at a section, and $A_{sv} = 2A_t$; f_c' denotes the compressive strength of concrete. For the machine learning models, the same training data is used to develop the predictive models, and the same test data is employed to evaluate the effectiveness of trained models. For the codes and empirical formula, all the data is included for performance evaluation.

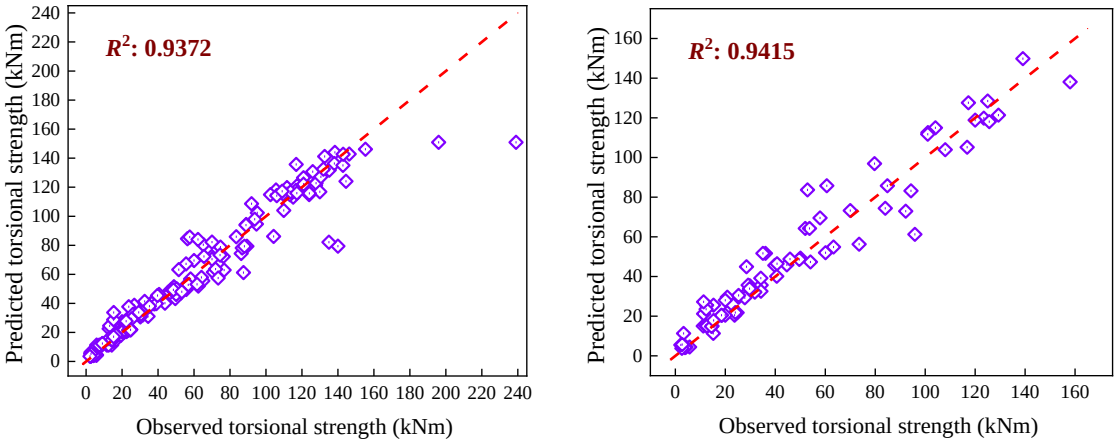
Figs. 10 to 13 present the comparisons between real torsional strength and outputs of four machine learning models for training and test data, respectively. According to the comparison results, it is noted that all the data-driven models have R-squared values above 0.9, which indicates acceptable torsional strength modeling capacity. Among four models, SVM has the worst performance in predicting torsional strength of RC beams, with corresponding R-squared values of 0.9074 and 0.9132 for training and test data, respectively. The GPR performs better than BT and 2D-CNN, with the coefficients of determination of 0.9863 and 0.9584, respectively. However, it is still inferior to the proposed 2D-CNN optimised by IBSA in terms of coefficient of determination. Fig. 14 shows the correlation analysis between real torsional strength and predictions from ACI-318-2005, BS-8110 and empirical formula proposed by Rahal, respectively, where the entire dataset is employed for the comparison. Obviously, building codes and empirical formula produce accurate predictions, when the torsional strength is no more than 40kNm. With the increase of torsional strength, the capabilities of these formulae gradually decline. Overall, the performance of ACI-318-2005 and Rahal proposed formula is satisfactory, with the coefficient of determination above 0.9. The BS-8110, however, performs worst among all the models and formulae, with the coefficient of determination of only 0.8052. Hence, from the point of view of correlation analysis, the proposed 2D CNN model with optimal hyperparameters exhibits the best performance in torsional capacity evaluation of RC beams.





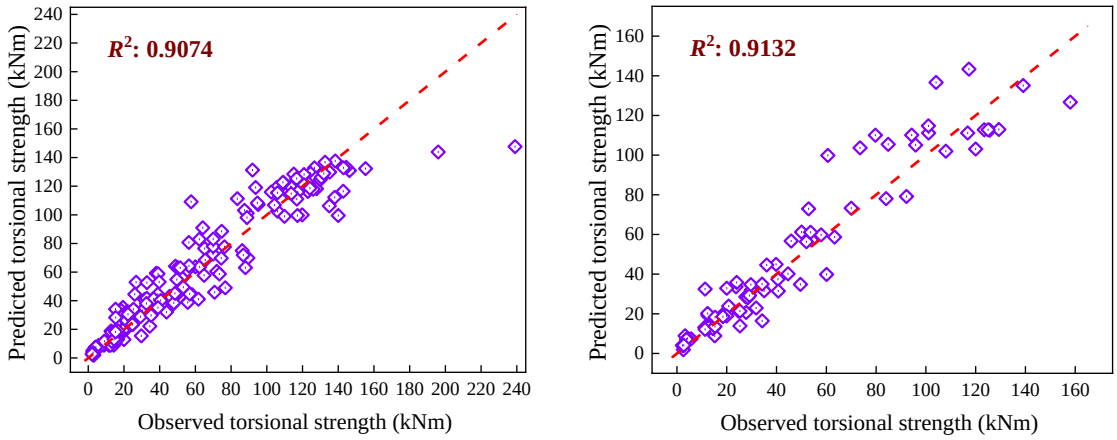
(a) Training (b) Test

Fig. 10. Comparison between real torsional strength and predictions of GPR model



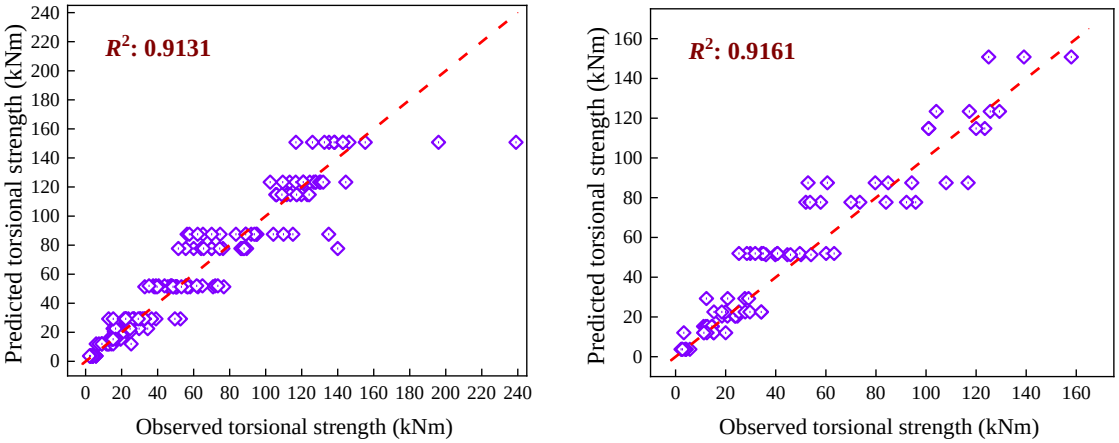
(a) Training (b) Test

Fig. 11. Comparison between real torsional strength and predictions of BT model



(a) Training (b) Test

Fig. 12. Comparison between real torsional strength and predictions of SVM model



(a) Training (b) Test

Fig. 13. Comparison between real torsional strength and predictions of CNN model

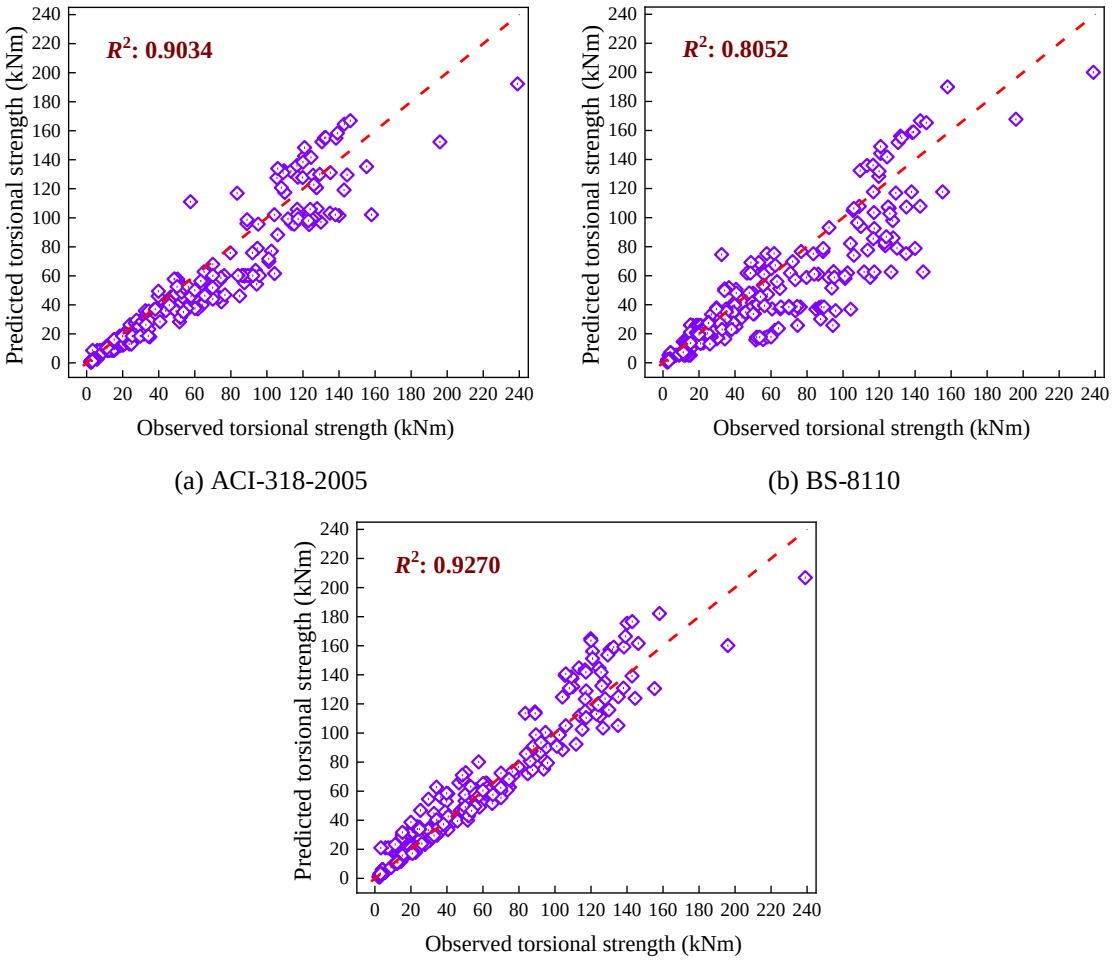


Fig. 14. Comparison between real torsional strength and predictions of building codes and empirical formula.

Fig. 15 summarised the distributions of absolute errors between real torsional strength and predictions of all the trained models and design formulae by box-plots. It is observed that the mean and median values of absolute errors between real strength and predictions from proposed model, GPR, BT, SVM and Rahal proposed formula are close to 0, while the corresponding values of ACI-318-2005 and BS-8110 obviously deviate from 0. In addition, the proposed model and GPR have relatively narrow ranges of absolute errors, indicating high prediction accuracy of trained models. The proposed model is superior to GPR due to the shorter range between lower and upper error bounds. Other models and formulae, however, indicate relatively poor accuracy in predicting the torsional strength, featured by wider outlier ranges.

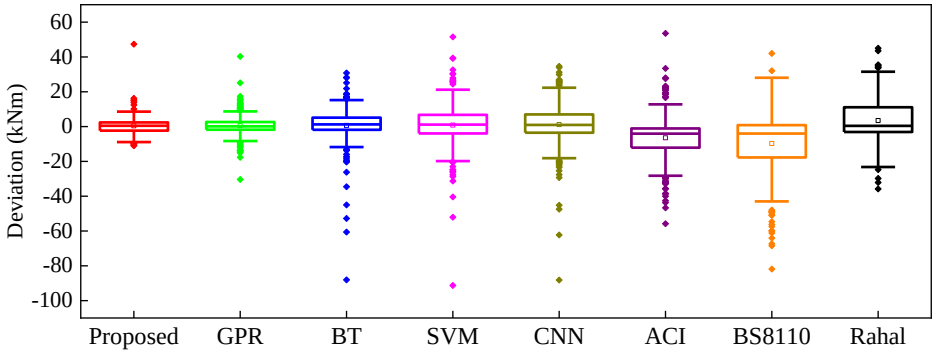


Fig. 15. Distributions of absolute error between real values and predictions of all models.

In addition to R-squared and absolute error distribution, more statistical evaluation metrics, such as mean absolute percentage error (MAPE), relative root mean square error (RRMSE), mean absolute error (MAE), scatter index (SI) and index of agreement (IA), are adopted to conduct a comprehensive assessment on model performance. The expressions of these metrics are shown in Eq. (26) to Eq. (30).

$$MAPE = \frac{\sum_{i=1}^{N_d} \left| \frac{ts_p(i) - ts_o(i)}{ts_o(i)} \right|}{N_d} \tag{26}$$

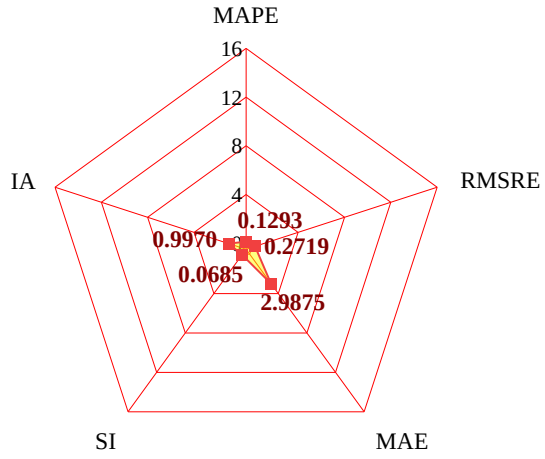
$$RMSRE = \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} \left[\frac{ts_p(i) - ts_o(i)}{ts_o(i)} \right]^2} \tag{27}$$

$$MAE = \frac{\sum_{i=1}^{N_d} \left| ts_p(i) - ts_o(i) \right|}{N_d} \tag{28}$$

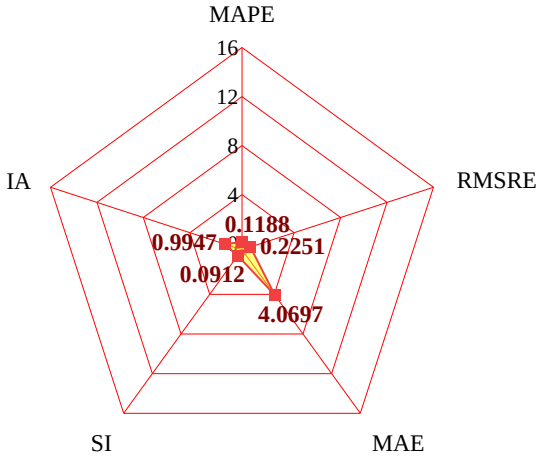
$$SI = \sqrt{\frac{\sum_{i=1}^{N_d} \left[\left(ts_p(i) - \frac{\sum_{i=1}^{N_d} ts_p(i)}{N_d} \right) - \left(ts_o(i) - \frac{\sum_{i=1}^{N_d} ts_o(i)}{N_d} \right) \right]^2}{\sum_{i=1}^{N_d} ts_o(i)^2}} \tag{29}$$

$$IA = 1 - \frac{\sum_{i=1}^{N_d} \left[ts_p(i) - ts_o(i) \right]^2}{\sum_{i=1}^{N_d} \left[\left| ts_p(i) - \frac{\sum_{i=1}^{N_d} ts_p(i)}{N_d} \right| + \left| ts_o(i) - \frac{\sum_{i=1}^{N_d} ts_o(i)}{N_d} \right| \right]^2} \tag{30}$$

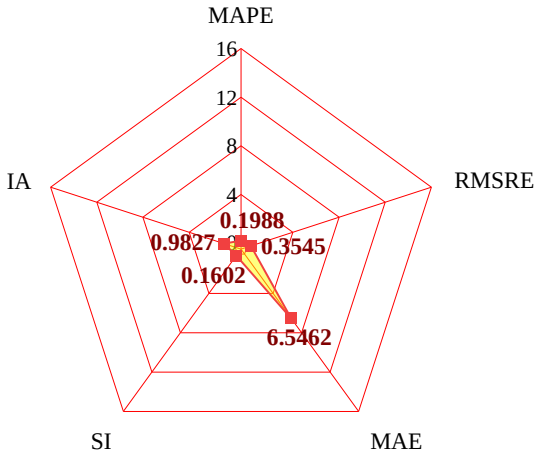
where N_d denotes the total sample number. For the metrics of MAPE, RMSRE, MAE and SI, the lower value indicates high capacity of model in predicting torsional strength of RC beams. For IA, however, a higher value corresponds to the better performance of the model. Fig. 16 illustrates the results of evaluation metrics of all the models based on the entire dataset in the form of radar diagrams. From the results, it is clearly seen that the proposed IBSA-2D-CNN possesses the optimal metrics of MAE, SI and IA, i.e., 2.9875, 0.0685 and 0.9970, respectively, while the GPR has the optimal metrics of MAPE and RMSRE, i.e., 0.1188 and 0.2251, respectively. For the metric of MAPE, the relative errors between proposed IBSA-2D-CNN model and other models/formulae are -8.18% for GPR, 53.70% for BT, 84.44% for SVM, 62.45% for 2D-CNN, 80.91% for ACI-318-2005, 124.24% for BS-8110 and 98.96% for Rahal formula, respectively; for the metric of RMSRE, the relative errors between proposed IBSA-2D-CNN model and other models/formulae are -17.23% for GPR, 30.38% for BT, 29.64% for SVM, 21.61% for 2D-CNN, 6.73% for ACI-318-2005, 30.21% for BS-8110 and 83.15% for Rahal formula, respectively; for the metric of MAE, the relative errors between proposed IBSA-2D-CNN model and other models/formulae are 36.22% for GPR, 119.12% for BT, 197.51% for SVM, 183.93% for 2D-CNN, 263.94% for ACI-318-2005, 407.13% for BS-8110 and 204.46% for Rahal formula, respectively; for the metric of SI, the relative errors between proposed IBSA-2D-CNN model and other models/formulae are 33.20% for GPR, 133.88% for BT, 181.39% for SVM, 174.66% for 2D-CNN, 189.65% for ACI-318-2005, 319.19% for BS-8110 and 161.25% for Rahal formula, respectively; for the metric of IA, the relative errors between proposed IBSA-2D-CNN model and other models/formulae are -0.24% for GPR, -1.44% for BT, -2.19% for SVM, -2.03% for 2D-CNN, -2.81% for ACI-318-2005, -6.86% for BS-8110 and -1.79% for Rahal formula, respectively. Overall, the proposed 2D-CNN with optimised hyperparameters outperforms other models and design formulae in torsional strength prediction of RC beams after a comprehensive evaluation, which can be considered as an effective method used in the design of RC beam.



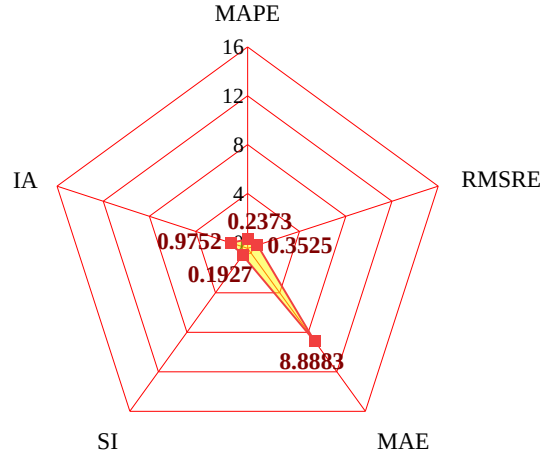
(a) Proposed



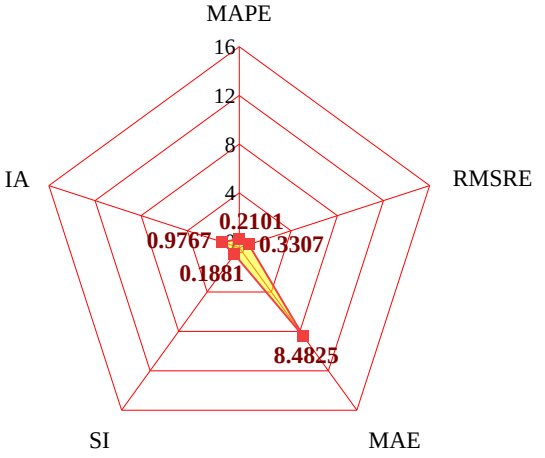
(b) GPR



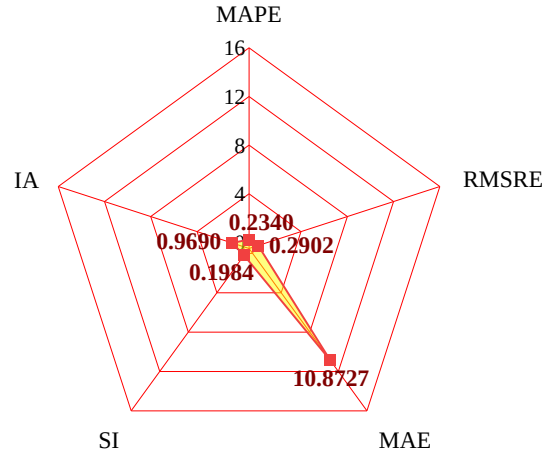
(c) BT



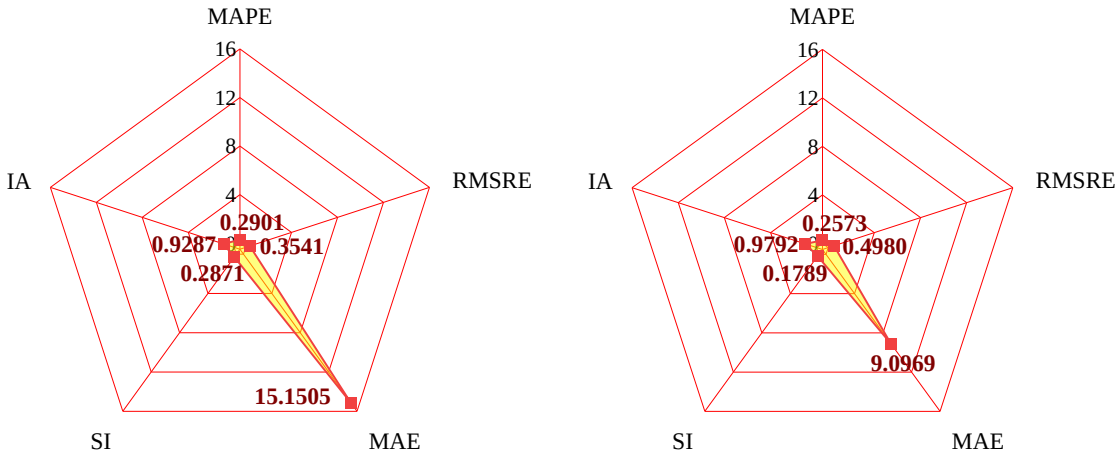
(d) SVM



(e) CNN



(f) ACI-318-2005



(g) BS-8110 (h) Rahal formula

Fig. 16. Evaluation results of eight models of torsional strength prediction of RC beams.

4.3. Sensitivity analysis of model inputs

The proposed drive-driven method has been proved to be effective in predicting torsional strength of RC beams with satisfactory performance. However, it belongs to the black box and has a complex inner structure of parameters without physical meaning, which prevents its practical application. Accordingly, to get the utmost out of the optimised 2D-CNN model for the design purpose of RC beam, it is necessary to study the sensitivity of each input variable on the network output. In this section, a numerical investigation is conducted to explore the nonlinear relationship between torsional strength output and each input variable. The reference values of inputs of optimised 2D-CNN model in this investigation are set as median values from the entire dataset, i.e., beam width sw = 254 mm, beam height sh = 381 mm, stirrup width stw = 216 mm, stirrup height sth = 327 mm, concrete compressive strength sc = 40.25 MPa, longitudinal reinforcement ratio lrr =1%, longitudinal reinforcement yield strength syt = 400 MPa, transverse reinforcement ratio tra = 0.91%, transverse reinforcement yield strength tyt = 385 MPa, and stirrup spacing sts = 92 mm. Then, each model input changes in a certain range from 50% to 150% of its reference value with the increment of 10%, while the other inputs remain constant in the meantime. Here, the relative error between model output and reference torsional strength is employed as a metric to measure the sensitivity of corresponding model input. The analysis results of input sensitivities of proposed IBSA optimised 2D-CNN for torsional strength prediction are displayed in Fig. 17, where a spide chart is drawn to indicate the error change from reference of each input variable. It is clearly seen that with the addition of input change percentage, the relative error between model output and reference value becomes increasingly large for all the variables. The mean value of relative errors of each parameter is summarised in Fig. 18 via a bar chart. It is noted that the parameters of beam height, stirrup width and height, with the averaged relative error values of 26.59%, 23.70% and 27.22%, respectively, are the three most sensitive parameters that significantly influence the output of proposed data-driven model. The parameters of transverse reinforcement ratio and transverse reinforcement yield strength, with the responding metric values of 2.97% and 3.28%, are the most insensitive parameters for the proposed model, which contribute less to the model output. The insensitive variables can be removed from the model inputs via constant assignments to simplify the network configuration, which is beneficial to the model training as well as practical applications via reducing the measurement costs.

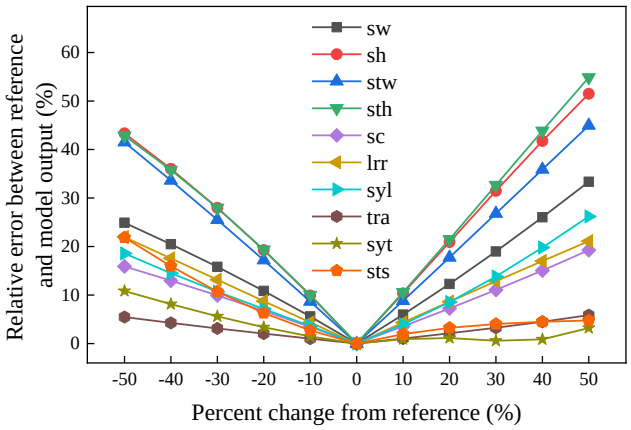


Fig. 17. Model input sensitivity analysis.

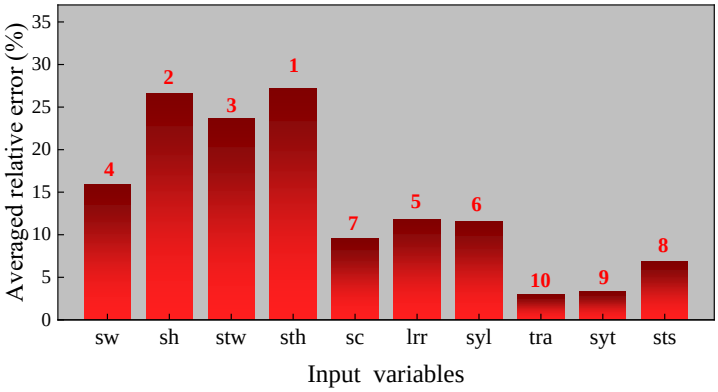


Fig. 18. Sensitivity ranking of different input variables.

5. Graphical user interface development for torsional capacity evaluation of RC beams

In the real application, the end-users of the developed 2D-CNN model with metaheuristic optimisation for predicting torsional strength of RC beams are structural engineers, who perhaps are not up on the fundamental of adopted technologies in this study. Accordingly, it is not easy to persuade them to use the proposed analytic tool for the design of RC beams. To address this concern, a practical and robust graphical user interface (GUI) is developed in this section based on Matlab App Designer to automate the analysis procedure, which is shown in Fig. 19. Based on this innovative interface, the structural designers just need to import the numerical values of ten model inputs and select the network type. Then, the predicted value of torsional strength of RC beam will be shown after the predict button is clicked. To clarify the use of this screening tool, a case study is presented as follows. Suppose the measurements of beam width, beam height, stirrup width, stirrup height, concrete compressive strength, longitudinal reinforcement ratio, longitudinal reinforcement yield strength, transverse reinforcement ratio, transverse reinforcement yield strength and stirrup spacing are 300 mm, 400 mm, 230 mm, 330 mm, 90 MPa, 0.993%, 529 MPa, 1.003%, 538 MPa and 70 mm, respectively. The CNN with hyperparameters optimised by IBSA is chosen, and the predicted value of torsional strength of RC beam is 84.77 kNm.

It should be noted that the proposed model is capable of providing accurate predictions of torsional strength, when the values of input parameters fall within the ranges of dataset used in this research. When the developed interface is used in practice, this limitation should be kept in mind to avoid the outrageous estimation. In the future, the dataset will be extended via more experimental data of RC beams, and the analytical model is then updated based on extended dataset to improve the estimation accuracy for larger input parameter ranges. Furthermore, an online training module will be added to the interface to update the developed model more conveniently.

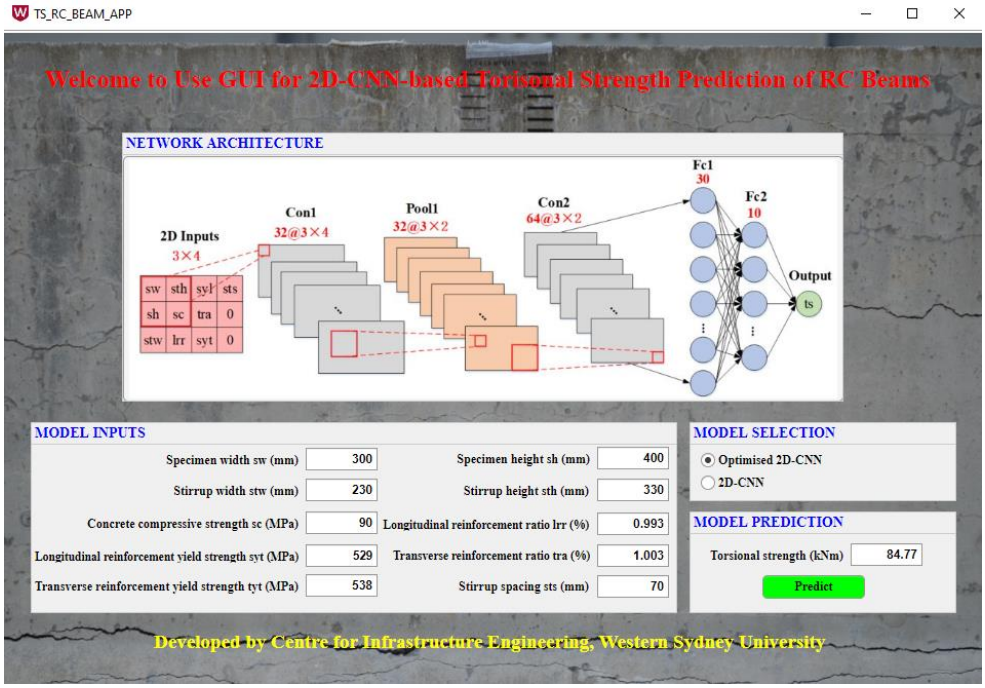


Fig. 19. GUI for torsional strength of RC beam using proposed method.

6. Conclusions

The lack of adequate torsional resistance capability of structural components can cause the appearance of great stress, which may further result in structural failure. Hence, this research investigates the evaluation of torsional capacity of RC beams by proposing a novel approach, where a data-driven model based on 2D-CNN is established. To improve the prediction performance of the proposed model, the IBSA is employed for the model hyperparameter optimisation. Finally, a comprehensive dataset consisting of 268 groups of experimental data of RC beams is used to train and test the proposed method. The significant outcomes of this research are summarised as follows.

- IBSA has been proved to be more efficient in optimising 2D-CNN hyperparameters for predicting torsional strength than BSA, PSO and GA due to higher accuracy and faster convergence.
- The proposed IBSA optimised 2D-CNN exhibits outstanding performance in torsional capacity evaluation of RC beams. It outperforms other machine learning models, building codes and empirical formula by means of a group of evaluation metrics.
- The sensitivity analysis results show that the beam width, beam height, stirrup width and stirrup height are the most sensitive parameters that significantly affect the model output, while the steel ratio of transverse reinforcement and yield strength of transverse reinforcement are insensitive parameters that could be excluded in the further development.
- Based on the proposed model, a GUI is designed to assist the structural engineers in the design of RC beams.

Acknowledgment

The authors would like to thank the support from Australian Research Council (Grant No. IH150100006) for conducting this research.

References

[1] Hsu TT. Torsion of structural concrete-behavior of reinforced concrete rectangular members. Special Publication. 1968;18:261-306.
[2] Chiu H-J, Fang I-K, Young W-T, Shiau J-K. Behavior of reinforced concrete beams with minimum torsional reinforcement. Engineering Structures. 2007;29:2193-205.
[3] Chalioris CE, Karayannis CG. Experimental investigation of RC beams with rectangular spiral reinforcement in torsion. Engineering structures. 2013;56:286-97.

[4] Rahal KN. Torsional strength of normal and high strength reinforced concrete beams. Engineering Structures. 2013;56:2206-16.
[5] Fiore A, Berardi L, Marano GC. Predicting torsional strength of RC beams by using evolutionary polynomial regression. Advances in Engineering Software. 2012;47:178-87.
[6] Arslan MH. Predicting of torsional strength of RC beams by using different artificial neural network algorithms and building codes. Advances in Engineering Software. 2010;41:946-55.
[7] Ilkhani M, Naderpour H, Kheyroddin A. A proposed novel approach for torsional strength prediction of RC beams. Journal of Building Engineering. 2019;25:100810.
[8] Haroon M, Koo S, Shin D, Kim C. Torsional Behavior Evaluation of Reinforced Concrete Beams Using Artificial Neural Network. Applied Sciences. 2021;11:4465.
[9] Han X-F, Huang X-Y, Sun S-J, Wang M-J. 3DDACNN: 3D dense attention convolutional neural network for point cloud based object recognition. Artificial Intelligence Review. 2022:1-17.
[10] Yaseen MU, Anjum A, Fortino G, Liotta A, Hussain A. Cloud based scalable object recognition from video streams using orientation fusion and convolutional neural networks. Pattern Recognition. 2022;121:108207.
[11] Pandey M, Xu Z, Sholle E, Maliakal G, Singh G, Fatima Z et al. Extraction of radiographic findings from unstructured thoracoabdominal computed tomography reports using convolutional neural network based natural language processing. PloS one. 2020;15:e0236827.
[12] Singhal N, Soni S, Bonthu S, Chattopadhyay N, Samanta P, Joshi U et al. A deep learning system for prostate cancer diagnosis and grading in whole slide images of core needle biopsies. Scientific reports. 2022;12:1-11.
[13] Gimenez M, Palanca J, Botti V. Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis. Neurocomputing. 2020;378:315-23.
[14] Feng L, Natu S, Som de Cerff Edmonds V, Valenza JJ. Multiphase flow detection with photonic crystals and deep learning. Nature Communications. 2022;13:1-10.
[15] Yu Y, Rashidi M, Samali B, Mohammadi M, Nguyen TN, Zhou X. Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. Structural Health Monitoring. 2022:14759217211053546.
[16] Liu Y, Xiao H, Xu J, Zhao J. A Rail Surface Defect Detection Method Based on Pyramid Feature and Lightweight Convolutional Neural Network. IEEE Transactions on Instrumentation and Measurement. 2022.
[17] 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.
[18] Deng F, He Y, Zhou S, Yu Y, Cheng H, Wu X. Compressive strength prediction of recycled concrete based on deep learning. Construction and Building Materials. 2018;175:562-9.
[19] Pathirage CSN, Li J, Li L, Hao H, Liu W, Wang R. Development and application of a deep learning–based sparse autoencoder framework for structural damage identification. Structural Health Monitoring. 2019;18:103-22.
[20] Fan G, Li J, Hao H, Xin Y. Data driven structural dynamic response reconstruction using segment based generative adversarial networks. Engineering Structures. 2021;234:111970.
[21] Jo J, Jadidi Z. A high precision crack classification system using multi-layered image processing and deep belief learning. Structure and infrastructure engineering. 2020;16:297-305.
[22] Yu Y, Wang C, Gu X, Li J. A novel deep learning-based method for damage identification of smart building structures. Structural Health Monitoring. 2019;18:143-63.
[23] Meng X-B, Gao XZ, Lu L, Liu Y, Zhang H. A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. Journal of Experimental & Theoretical Artificial Intelligence. 2016;28:673-87.
[24] Wu D, Pun C-M, Xu B, Gao H, Wu Z. Vehicle power train optimization using multi-objective bird swarm algorithm. Multimedia Tools and Applications. 2020;79:14319-39.
[25] Alhassan AM, Wan Zainon WMN. Taylor bird swarm algorithm based on deep belief network for heart disease diagnosis. Applied Sciences. 2020;10:6626.
[26] Pruthi J, Arora S, Khanna K. Modified Bird swarm algorithm for edge detection in noisy images using fuzzy reasoning. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization. 2018.
[27] Xiang L, Deng Z, Hu A. Forecasting short-term wind speed based on IEWT-LSSVM model optimized by bird swarm algorithm. IEEE Access. 2019;7:59333-45.
[28] Malathy E, Asaithambi M, Dheeraj A, Arputharaj K. Hybrid Bird Swarm Optimized Quasi Affine Algorithm Based Node Location in Wireless Sensor Networks. Wireless Personal Communications. 2021:1-16.
[29] Niu M, Ma S, Cai W, Zhang J, Liu G. Fault Diagnosis of Planetary Roller Screw Mechanism Based on Bird Swarm Algorithm and Support Vector Machine. Journal of Physics: Conference Series: IOP Publishing; 2020. p. 012007.
[30] Lee J-Y, Kim S-W. Torsional strength of RC beams considering tension stiffening effect. Journal of structural engineering. 2010;136:1367-78.

[31] Kim C, Kim S, Kim K-H, Shin D, Haroon M, Lee J-Y. Torsional Behavior of Reinforced Concrete Beams with High-Strength Steel Bars. ACI Structural Journal. 2019;116.
[32] Rahal KN, Collins MP. Simple model for predicting torsional strength of reinforced and prestressed concrete sections. Structural journal. 1996;93:658-66.
[33] Rasmussen L, Baker G. Torsion in reinforced normal and high-strength concrete beams part 1: experimental test series. Structural Journal. 1995;92:56-62.
[34] Lee J-Y, Kim K-H, Lee SH, Kim C, Kim M-H. Maximum Torsional Reinforcement of Reinforced Concrete Beams Subjected to Pure Torsion. ACI Structural Journal. 2018;115.
[35] Fang I-K, Shiau J-K. Torsional behavior of normal-and high-strength concrete beams. Structural Journal. 2004;101:304-13.
[36] Koutchoukali N-E, Belarbi A. Torsion of high-strength reinforced concrete beams and minimum reinforcement requirement. Structural Journal. 2001;98:462-9.
[37] McMullen AE, Rangan BV. Pure Tension in Rectangular Sections-A Re-Examination. Journal Proceedings1978. p. 511-9.
[38] Kim M-J, Kim H-G, Lee Y-J, Kim D-H, Lee J-Y, Kim K-H. Pure torsional behavior of RC beams in relation to the amount of torsional reinforcement and cross-sectional properties. Construction and Building Materials. 2020;260:119801.
[39] Joh C, Kwahk I, Lee J, Yang I-H, Kim B-S. Torsional behavior of high-strength concrete beams with minimum reinforcement ratio. Advances in Civil Engineering. 2019;2019.
[40] Al-Faqra E, Murad Y, Abdel Jaber Mt, Shatarat N. Torsional behaviour of high strength concrete beams with spiral reinforcement. Australian Journal of Structural Engineering. 2021;22:266-76.
[41] Victor DJ, Muthukrishnan R. Effect of stirrups on ultimate torque of reinforced concrete beams. Journal Proceedings1973. p. 300-6.
[42] Tang C-W. Using radial basis function neural networks to model torsional strength of reinforced concrete beams. Computers and Concrete. 2006;3:335.
[43] Zhang Y. Torsion in high-strength concrete rectangular beams: University of Nevada, Reno; 2002.
[44] Peng X-N, Wong Y-L. Behavior of reinforced concrete walls subjected to monotonic pure torsion—An experimental study. Engineering structures. 2011;33:2495-508.
[45] Li S, Zhao X. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. Advances in Civil Engineering. 2019;2019.
[46] Code A. Building code requirements for structural concrete (ACI 318-05) and commentary (ACI 318R-05). American Concrete Institute, Farmington Hills, Mich. 2005.
[47] Standard B. Structural use of concrete. BS8110. 1986.