Computers and Concrete, Vol. 3, No. 5 (2006) 335-355
DOI: http://dx.doi.org/10.12989/cac.2006.3.5.335

Using radial basis function neural networks to model torsional strength of reinforced concrete beams

Chao-Wei Tang†

Department of Civil Engineering, Cheng-Shiu University,
No. 840, Chengcing Road, Niaosong Township, Kaohsiung County, Taiwan, R.O.C.

(Received July 7, 2006, Accepted September 15, 2006)

Abstract. The application of radial basis function neural networks (RBFN) to predict the ultimate torsional strength of reinforced concrete (RC) beams is explored in this study. A database on torsional failure of RC beams with rectangular section subjected to pure torsion was retrieved from past experiments in the literature; several RBFN models are sequentially built, trained and tested. Then the ultimate torsional strength of each beam is determined from the developed RBFN models. In addition, the predictions of the RBFN models are also compared with those obtained using the ACI 318 Code equations. The study shows that the RBFN models give reasonable predictions of the ultimate torsional strength of RC beams. Moreover, the results also show that the RBFN models provide better accuracy than the existing ACI 318 equations for torsion, both in terms of root-mean-square error and coefficients of determination.

Keywords: reinforced concrete beam; torsional strength; radial basis function network.

1. Introduction

Numerous analytical models have been reported on the torsional behavior of reinforced concrete (RC) beams subjected to pure torsion or combined torsion, bending, and shear since the 1920s. Thus the torsional behavior of RC beams are able to be predicted accurately, including the torque-twist curve, stiffness, cracking state, yielding state, ultimate state, and descending branch. Although the characteristic parameters for these models have been carefully examined, their implementation into design codes still requires considerable simplification. Recent researches have shown that artificial neural network-based modeling is an alternative method for modeling complex nonlinear relationship. An artificial neural network (ANN) is simply a computational tool that attempts to simulate the architecture and internal features of the human brain and nervous system (Sanad and Saka 2001). Much of the success of ANN is due to such characteristics as nonlinear processing and parallel processing. Moreover, it can learn functional relationships from examples without prior knowledge of the underlying mathematical model and thus discover patterns and regularities in data through self-organization (Hopfield 1982). In civil engineering, the methodology of ANN has been successfully applied to model the structural behavior and properties of concrete materials such as strength and constitutive modeling (Hajela and Berke 1991, Consolazio 2000, Tang, *et al*. 2003, Ghaboussi*, et al.* 1991, Yeh 1999, Zhao and Ren 2002).

The most commonly used ANN is probably the multilayer perceptrons (MLP) network with back-

† Associate Professor, E-mail: tangcw@csu.edu.tw

propagation algorithm that uses the gradient-descent method to minimize the error between the network outputs and the target outputs (Rumelhart, *et al.* 1986). This type of neural network is known as a supervised network because it requires a target output in order to learn. And it has been proved a powerful data-modeling tool that is able to capture and represent complex input/output relationships. However, for nonlinear modeling themes in real applications, MLP networks have poor process interpretability and are hindered by problems associated with weight optimization such as slow learning and local minimization (Jang, *et al.* 1997). Recently, radial basis function networks (RBFNs) have been applied as alternatives to alleviate some of the limitations of MLP networks. The training of RBFNs can be split into an unsupervised part and a supervised but linear part. Unsupervised updating techniques are straightforward and relatively fast. Meanwhile its supervised part consists in solving a linear problem, which is therefore also fast. The training methods used for RBFNs are thus substantially less time and resources consuming (Bishop 1995). Therefore, RBFNs have gradually become one of the most popular feedforward neural networks with applications in regression, classification and function approximation problems (Bishop 1995, Haykin 1994).

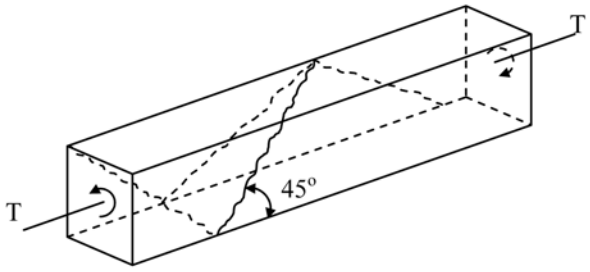
In light of the availability of more experimental data and recent advance in the area of data analysis techniques, it would be significant to develop new methods that are easier, convenient, and accurate than the existing methods. Based on the ANNs technology, this paper presents a nontraditional approach to the prediction of the ultimate torsional strength of RC beams with rectangular section subjected to pure torsion. In this study a commercially available software package, STATISTICA Neural Networks, was used to establish the RBFN models. A database of 76 records including normal- and high-strength concretes was retrieved from the existing literature for analysis. Besides, a comparative study between the developed RBFN models and the ACI Building Code equations for torsion strength is also made. The findings will provide valuable information for modeling the torsional strength of RC beams.

2. Prediction of torsional strength

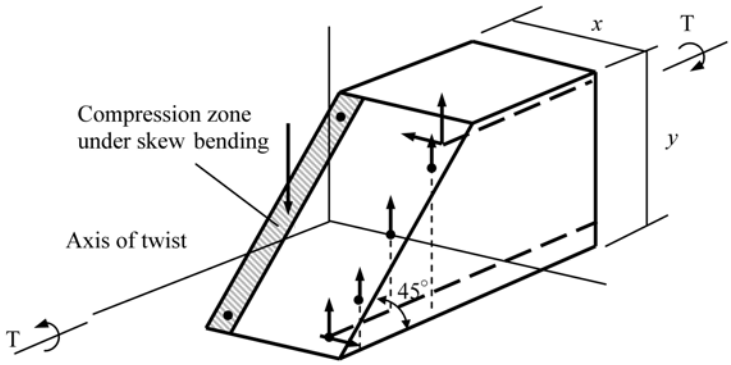
The method for analysis of torsional strength can be roughly classified into two main categories: the skew-bending and the space-truss analogy theory. In this section, the two theories for torsional strength of reinforced concrete members are reviewed briefly. On the other hand, the ACI Building Code provisions for torsional design were selected and used in this study for comparison with the results from the RBFN models. Therefore, the ACI equations for torsional strength of RC beams are also outlined in the following.

2.1. Skew-bending theory

The skew-bending theory considers in detail the internal deformation behavior of the series of transverse warped surfaces along the beam (Lessig 1958). Its basic aspect is the assumption of a skew failure surface that is initiated by a helical crack on the three faces of a rectangular beam while compression zone near the fourth face connects the ends of this helical crack, as shown in Fig. 1. Afterward, it was further developed and applied to concrete structures (Yudin 1962, Collion, *et al.* 1965, Goode and Helmy 1968, Hsu 1968a). And thus this theory formed the basis for the 1971 ACI Building Code provisions for torsional design of RC beams subjected to combined actions of torsion, bending, and shear.



(a) Skew-bending due to torsion



(b) Forces acting on skew plane at failure

Fig. 1 Skew-bending theory analogy

Essentially, the 1971 ACI Building Code provisions for torsion design of RC beams remain in the ACI 318-89 Building Code (1989). Therefore, the ACI 318-89 Building Code specifies the nominal torsional strength T_n of reinforced concrete members was contributed by both concrete T_c and torsional reinforcements T_s and expressed as

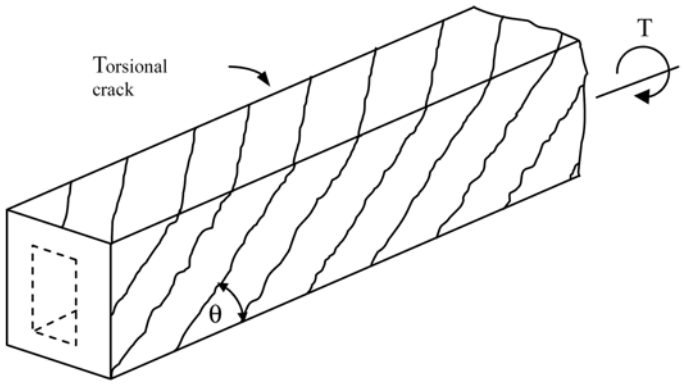
$$T_n = T_c + T_s = 0.067x^2y\sqrt{f'_c} + \alpha_t \frac{x_1 y_1 A_t f_{yv}}{s} \text{(SI units)} \quad (1)$$

where

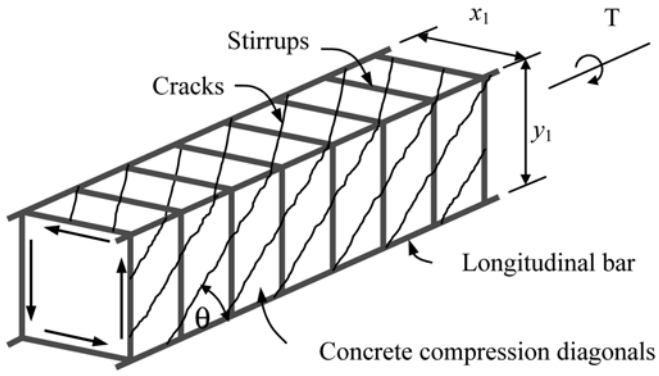
x = short dimension of the cross section;
y = long dimension of the cross section;
f'_c = concrete compressive strength, MPa;
f_{yv} = yield strength of closed stirrups, MPa;
A_t = cross-sectional area of one leg of closed stirrup;
s = spacing of stirrups;
α_t = $0.66+0.33(y_1/x_1)\leq 1.5$;
x_1 = short dimension of the closed stirrup; and
y_1 = long dimension of the closed stirrup.

2.2. Space-truss analogy theory

In fact, the space-truss analogy was the first theory for calculating the torsional strength of reinforced concrete members subjected to torsion. It was initially proposed by Rausch (1929) and further developed by many scholars in this field (Anderson 1935, Cowan 1950, Elfegren, *et al.* 1974, Hsu and Mo 1985a, Hsu and Mo 1985b, MacGregor and Ghoneim 1995). It is assumed in this theory that the concrete beam behaves in torsion similar to a thin-walled box with a constant shear flow in the wall cross-section, producing a constant torsional moment. This assumption was proved to be true since later experiments have shown that the ultimate torques of hollow and solid members are virtually the same. In other words, once cracking has occurred, the concrete in the center of the member has little effect on the torsional strength of the cross section and can be ignored. Accordingly, in the process of torsion design for a RC beam, the beam can be considered to be equivalent tubular member, as shown in Fig. 2 (MacGregor and Ghoneim 1995). After cracking, the tube is idealized as a space truss consisting of closed stirrups, longitudinal steel bars in the corners, and concrete compression diagonals approximately centered on the stirrups. The diagonals are at an angle θ to the member longitudinal axis, as shown in Fig. 2. Generally, the



(a) Thin-walled tube analogy



(b) Space-truss analogy

Fig. 2 Thin-walled tube and space-truss analogy

design method for torsion based on the thin-walled tube, space-truss analogy is considerably simpler to understand and apply, and is equally accurate. Consequently, since 1995, the torsion provisions in the ACI 318 Code have been substantially revised using the thin-walled tube analogy.

According to the current torsion provision of ACI 318-05 (2005), the nominal torsional strength T_n of RC beams is assumed to be resisted by the closed stirrups and longitudinal steel while the torsion moment T_c resisted by the concrete compression struts is assumed as zero. Therefore, the following equation is given for T_n

$$T_n = \left(\frac{2A_0 A_t f_{yv}}{s}\right)\cot\theta \tag{2}$$

in which

$$\cot\theta = \sqrt{\frac{A_l f_{yl} s}{A_t f_{yv} p_h}} \tag{3}$$

where

A_0 = gross area enclosed by shear flow path;
θ = angle of compression diagonals;
f_{yl} = yield strength of longitudinal torsional reinforcement, MPa;
A_l = total area of longitudinal torsional reinforcement; and
p_h = perimeter of centerline of outmost closed transverse torsional reinforcement.

3. Radial basis function networks

A RBFN is a special kind of neural network, which contains three layers with different functional roles: the input, hidden and output layers. The input layer serves only as input distributor to the hidden layer, where they are transformed into a high-dimensional feature space using radial basis functions, called basis functions. The basis functions are exponentially decaying nonlinear functions and are radial functions, which have radial symmetry with respect to a center. The centers can be regarded as the neurons (or nodes) of the hidden layer. Each neuron in the hidden layer is a radial function. Then, the transformed data in this space are linearly transformed in order to approximate the target outputs. The architecture and training algorithm of RBFNs are described below.

3.1. Architecture of RBFNs

The problem of interpolation of real multivariable functions can be expressed as follows. Given a set of N data points in the input space R^D, together with their associate target output values in the output space R^P, one has to calculate a function $f(\mathbf{x})$ from R^D to R^P. In a RBFN, $f(\mathbf{x})$ is approximated by a set of D-dimensional radial activation functions. Fig. 3 shows the typical architecture of a RBFN with an input layer of D neurons, a hidden layer of M neurons, an output layer of P neurons, adjustable weights that exist only between the hidden and output layers, and biases at each output neuron. Given a set of N data points in a multidimensional space, the aim of exact interpolation is to find a function such that every D-dimensional input feature vector $\mathbf{x}^n=\{x_i^n: i=1,\ldots, D\}$ has a corresponding P-dimensional target output vector $\mathbf{f}^n = \{f_k^n: k=1,\ldots, P\}$. The approximation of function $f(\mathbf{x})$ may be expressed as a linear combination of the radial basis functions,

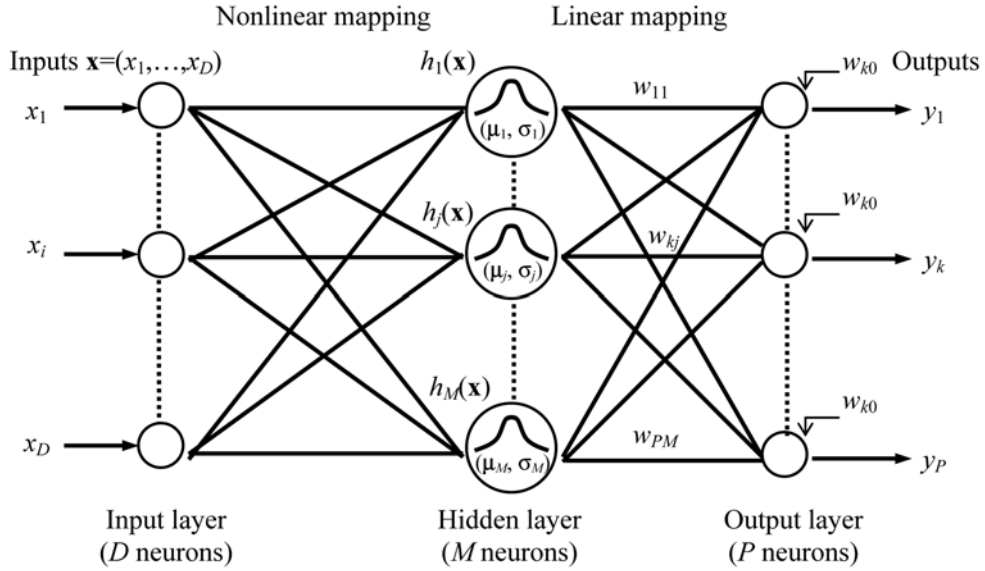


Fig. 3 Architecture of a typical RBFN

of which the kth output of the network consist of sums of the weighted hidden layer neurons plus the bias when the network is presented with the nth input vector and can be expressed by

$$\hat{f}_k(\mathbf{x}^n) = \sum_{j=1}^{M} w_{kj} h_j(\mathbf{x}^n) + w_{k0}, \qquad k = 1, 2, \ldots, P \tag{4}$$

where

w_{kj} = weight connecting the jth basis function and the kth output;
$h_j(\mathbf{x}^n)$ = output from the jth hidden neuron for the input vector $\mathbf{x}^n$; and
w_{k0} = a bias term at the kth output neuron.

A typical choice for the radial basis functions is a set of multi-dimensional Gaussian kernel, its dimensionality being the same as the D-dimensional input vector $\mathbf{x}^n$. Thus, the output $h_j(\mathbf{x}^n)$ from the jth hidden neuron for the input vector $\mathbf{x}^n$ has the following form:

$$h_j(\mathbf{x}^n;\ \boldsymbol{\mu}_j,\ \sigma_j) = \exp\left[-\left(\frac{\|\mathbf{x}^n - \boldsymbol{\mu}_j\|}{\sqrt{2}\sigma_j}\right)^2\right], \qquad j = 1, 2, \ldots, M \tag{5}$$

where

$\boldsymbol{\mu}_j$ = a center vector with the same dimension as $\mathbf{x}^n$;
σ_j = width factor (or standard deviation) of the jth basis function; and
$\|\cdot\|$ = Euclidean distance between an input vector $\mathbf{x}^n$ and a center vector $\boldsymbol{\mu}_j$.

The output $h_j(\mathbf{x}^n)$ has a significant response to the input vector only over a limited range called the receptive field of which the size is determined by the value of σ. The receptive field is controlled by the center $\boldsymbol{\mu}_j$ and the width σ_j of the neuron, i.e., the shape of the exponential function. The receptive field is that region in space over which the neuron has appreciable response. It is restricted to a small region of the input space. This ensures that there exists a set of weights that approximates the target function better than any other set.

Once the general shape of the h_j function is chosen, the purpose of a RBFN algorithm is to find the parameters $\boldsymbol{\mu}_j$, σ_j; and w_{kj} to best fit function $f(\mathbf{x})$. Fitting means here that the global mean-square error between the target output $\mathbf{f}^n = \{f_k^n : k = 1, \ldots, P\}$ for all N data points and the estimated output $f_k(\mathbf{x}^n)$ is minimized. As with MLPs the sum-of-squares error function is given by

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{P} (\hat{f}_k(\mathbf{x}^n) - f_k^n)^2 \tag{6}$$

3.2. Training algorithm of RBFNs

As pointed out previously, a RBFN learning algorithm consists of finding the parameters $\boldsymbol{\mu}_j$, σ_j; and w_{kj}. In practice, the training algorithm of a RBFN can be divided into a two-stage procedure: (1) determine the centers of the Gaussian kernels and compute their widths; and (2) compute the weights between the hidden and output layers. In the literature, a large variety of training algorithms have been tested in RBFNs. Most of these training algorithms correspond to supervised training or to a joint unsupervised-supervised paradigm. During the first stage, the position of the centers and the widths of the radial basis functions are obtained by unsupervised learning rules such as *K*-means clustering algorithm (Moody and Darken 1989) or supervised learning rules such as learning vector quantization (Kohonen 1988, Schwenker, *et al.* 1994). In the second stage, the weights of the output layer are trained using supervised methods, for example, direct least squares methods such as the Moore–Penrose pseudo-inverse method (Haykin 1999) or nonlinear iterative techniques such as a gradient descent method. Here, the *K*-means clustering algorithm, the pseudo-inverse algorithm, and the gradient descent method are briefly described in the following.

The *K*-means clustering algorithm is a nonhierarchical clustering algorithm, for which the number K of the data clusters basis functions has to be decided in advance, and then follows a simple re-estimation procedure to partition the data points $\mathbf{x}^n = \{\mathbf{x}_i^n : i = 1, \ldots, D\}$ into K disjoint subsets S_j containing N_j data points to minimize the sum squared clustering function, defined as:

$$J = \sum_{j=1}^{K}\sum_{n\in S_j} \left\| \mathbf{x}^n - \boldsymbol{\mu}_j \right\|^2 \tag{7}$$

where $\boldsymbol{\mu}_j$ is the mean/centroid of the data points in subset S_j given by

$$\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_{n\in S_j} \mathbf{x}^n \tag{8}$$

Once the centers of the K subsets are determined, the width factor of the radial basis functions has to be estimated. One of the simplest ways is to set the σ_j value equal to the Euclidean distance of the jth center $\boldsymbol{\mu}_j$ from its nearest neighbors or to take into account the R nearest neighbors (Moody and Darken 1989). Then:

$$\sigma_j = \frac{1}{R}\left(\sum_{i=1}^{R} \left\| \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \right\|^2\right)^{1/2} \tag{9}$$

where $\boldsymbol{\mu}_i$ are the R-nearest neighbors of centroid $\boldsymbol{\mu}_j$.

Provided that the number and shape of the radial basis functions in the receptive field have been determined, the weights of the output layer can be calculated. Considering the RBFN mapping

defined in Eq. (4) and absorbing the bias parameter into the weights. Then Eq. (4) can be expressed as the following equation

$$\hat{f}_k(\mathbf{x}^n) = \sum_{j=0}^{M} w_{kj} h_j(\mathbf{x}^n), \qquad k = 1, 2, \ldots, P \tag{10}$$

where h_0 is an extra RBF with activation value fixed at 1. Defining matrices with components $(W)_{kj} = w_{kj}$ the weight connecting the jth basis function and the kth output, $(H)_{nj} = h_j(\mathbf{x}^n)$ the outcome of the jth basis function with the nth feature vector $\mathbf{x}^n$ as input, and $(F)_{nk}=\{f_k^n\}$ the kth component of the nth target vector $\mathbf{f}^n$, respectively, the above equation can be written in matrix notation as

$$F = WH \tag{12}$$

For a large class of functions, the matrix H is non-singular, provided the data points are distinct. Then the network weights can be computed by fast linear matrix inversion techniques (Bishop 1995). And the formal solution for the matrix of the output layer weights W is given in the form

$$W = (H^T H)^{-1} H^T F = H^+ F$$

where $H^+=(H^T H)^{-1}H^T$ denotes the pseudo-inverse of H. It can be seen to have the property $H^+H= \mathbf{I}$. In practice, it is prefer to use singular value decomposition (SVD) to avoid possible ill-conditioning of H, i.e., H^+H being singular or near singular. The output layer weights can also be found by gradient descent optimization of the sum-of-squares error function defined in Eq. (6). In other words, the network is trained by adjusting its weights so as to minimise the error function over the entire training data set. This leads to the delta learning rule for the output weights. Hence, the change of the weight is given by

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \tag{13}$$

where η is the learning rate and $\partial E/\partial w_{kj}$ is the gradient. Then, the output layer weights would be iteratively updated using the following equation

$$w_{kj}^{old} = w_{kj}^{new} + \Delta w_{kj} \tag{14}$$

4. RBFN modeling of torsion strength

A commercially available software package, STATISTICA Neural Networks, was used to establish the RBFN models for predicting the ultimate torsional strength of RC beams. Details on the establishment of neural network-based models for torsion strength, along with sources of the data that are used in the development, are described below.

4.1. Data set and system model

The experimental data used in this study include 76 records retrieved from the existing literature (Fang and Shiau 2004, Hsu 1968b, Koutchoukali and Belarbi 2001, Rasmussen and Baker 1995). The complete list of the data is given in Table 1, where the name and the source of each specimen are referenced, and their ranges are listed in Table 2. The data set is then divided into three subsets;

Table 1 Experimental data

No.	x (mm)	y (mm)	x_1 (mm)	y_1 (mm)	f'_c (MPa)	s (mm)	A_t (mm^2)	f_{yv} (MPa)	A_l (mm^2)	f_{yl} (MPa)	ρ_t (%)	ρ_l (%)	T_u (kN-m)	Type of sub-set	Source Ref.
H-06-06	350	500	300	450	78.5	100	71.33	440	1191.6	440	0.61	0.68	92.0	test	Fang and Shiau (2004)
H-06-12	350	500	300	450	78.5	100	71.33	440	2027.2	410	0.61	1.16	115.1	verification	
H-12-12	350	500	300	450	78.5	50	71.33	440	2027.2	410	1.22	1.16	155.3	test	
H-12-16	350	500	300	450	78.5	50	71.33	440	2865	520	1.22	1.64	196.0	training	
H-20-20	350	500	300	450	78.5	55	126.7	440	3438	560	1.97	1.96	239.0	training	
H-07-10	350	500	300	450	68.4	90	71.33	420	1719	500	0.68	0.98	126.7	test	
H-14-10	350	500	300	450	68.4	80	126.7	360	1719	500	1.36	0.98	135.2	training	
H-07-16	350	500	300	450	68.4	90	71.33	420	2865	500	0.68	1.64	144.5	test	
N-06-06	350	500	300	450	35.5	100	71.33	440	1191.6	440	0.61	0.68	79.7	training	
N-06-12	350	500	300	450	35.5	100	71.33	440	2027.2	410	0.61	1.16	95.2	training	
N-12-12	350	500	300	450	35.5	50	71.33	440	2027.2	410	1.22	1.16	116.8	test	
N-12-16	350	500	300	450	35.5	50	71.33	440	2865	520	1.22	1.64	138.0	verification	
N-20-20	350	500	300	450	35.5	55	126.7	440	3438	560	1.97	1.96	158.0	verification	
N-07-10	350	500	300	450	35.5	90	71.33	420	1719	500	0.68	0.98	111.7	test	
N-14-10	350	500	300	450	35.5	80	126.7	360	1719	500	1.36	0.98	125.0	training	
N-07-16	350	500	300	450	35.5	90	71.33	420	2865	500	0.68	1.64	117.3	test	
B5UR1	203	305	165	267	39.6	108	71.33	373	506.8	386	0.92	0.82	19.4	test	Koutchoukali and Belarbi (2001)
B7UR1	203	305	165	267	64.6	108	71.33	399	506.8	386	0.92	0.82	18.9	verification	
B9UR1	203	305	165	267	75	108	71.33	373	506.8	386	0.92	0.82	21.1	verification	
B12UR1	203	305	165	267	80.6	108	71.33	399	506.8	386	0.92	0.82	19.4	verification	
B14UR1	203	305	165	267	93.9	108	71.33	386	506.8	386	0.92	0.82	21.0	training	
B12UR2	203	305	165	267	76.2	102	71.33	386	506.8	386	0.98	0.82	18.4	test	
B12UR3	203	305	165	267	72.9	95	71.33	386	649.46	373	1.05	1.05	22.5	training	
B12UR4	203	305	165	267	75.9	90	71.33	386	760.2	373	1.11	1.23	23.7	training	
B12UR5	203	305	165	267	76.7	70	71.33	386	794.4	380	1.42	1.28	24.0	training	

Note: The definitions of the parameters are shown in Table 2.

Table 1 Continued

No.	x (mm)	y (mm)	x_1 (mm)	y_1 (mm)	f'_c (MPa)	s (mm)	A_t (mm^2)	f_{yv} (MPa)	A_l (mm^2)	f_{yl} (MPa)	ρ_t (%)	ρ_l (%)	T_u (kN-m)	Type of subset	Source Ref.
B30.1	160	275	130	245	41.7	90	78.54	665	1543.9	620	1.49	3.51	16.6	test	Rasmussen and Baker (1995)
B30.2	160	275	130	245	38.2	90	78.54	669	1543.9	638	1.49	3.51	15.3	training	
B30.3B	160	275	130	245	36.3	90	78.54	672	1543.9	605	1.49	3.51	15.3	verification	
B50.1	160	275	130	245	61.8	90	78.54	665	1543.9	612	1.49	3.51	20.0	training	
B50.2	160	275	130	245	57.1	90	78.54	665	1543.9	614	1.49	3.51	18.5	test	
B50.3	160	275	130	245	61.7	90	78.54	665	1543.9	612	1.49	3.51	19.1	training	
B70.1	160	275	130	245	77.3	90	78.54	658	1543.9	617	1.49	3.51	20.1	test	
B70.2	160	275	130	245	76.9	90	78.54	656	1543.9	614	1.49	3.51	20.7	training	
B70.3	160	275	130	245	76.2	90	78.54	663	1543.9	617	1.49	3.51	21.0	training	
B110.1	160	275	130	245	109.8	90	78.54	655	1526.8	618	1.49	3.47	24.7	training	
B110.2	160	275	130	245	105	90	78.54	660	1526.8	634	1.49	3.47	23.6	training	
B110.3	160	275	130	245	105.1	90	78.54	655	1543.9	629	1.49	3.51	24.8	verification	
B1	254	381	215.9	342.9	27.58	152.4	71.33	341.29	508	313.71	0.54	0.52	22.3	test	Hsu (1968b)
B2	254	381	215.9	342.9	28.61	181.1	126.7	319.92	635	316.47	0.81	0.66	29.3	verification	
B3	254	381	215.9	342.9	28.06	127	126.7	319.92	762	327.5	1.15	0.79	37.5	training	
B4	254	381	215.9	342.9	30.54	92.2	126.7	323.36	889	319.92	1.59	0.92	47.3	test	
B5	254	381	215.9	342.9	29.03	69.9	126.7	321.3	1016	332.33	2.09	1.05	56.2	training	
B6	254	381	215.9	342.9	28.82	57.2	126.7	322.67	1143	331.64	2.56	1.18	61.7	training	
B7	254	381	215.9	342.9	25.99	127	126.7	318.54	508	319.92	1.15	0.52	26.9	training	
B8	254	381	215.9	342.9	26.75	57.2	126.7	319.92	508	321.99	2.56	0.52	32.5	training	
B9	254	381	215.9	342.9	28.82	152.4	126.7	342.67	762	319.23	0.96	0.79	29.8	training	
B10	254	381	215.9	342.9	26.48	152.4	126.7	341.98	1143	334.4	0.96	1.18	34.4	verification	
D1	254	381	215.9	342.9	26.61	152.4	71.33	337.84	508	333.02	0.54	0.52	22.4	training	
D2	254	381	215.9	342.9	25.58	181.1	126.7	330.95	635	322.67	0.81	0.66	27.7	verification	
D3	254	381	215.9	342.9	28.41	127	126.7	333.02	762	341.29	1.15	0.79	40.2	training	
D4	254	381	215.9	342.9	30.61	92.2	126.7	333.02	889	330.26	1.59	0.92	47.9	training	

Note: The definitions of the parameters are shown in Table 2.

Table 1 Continued

No.	x (mm)	y (mm)	x_1 (mm)	y_1 (mm)	f'_c (MPa)	s (mm)	A_t (mm^2)	f_{yv} (MPa)	A_l (mm^2)	f_{yl} (MPa)	ρ_t (%)	ρ_l (%)	T_u (kN-m)	Type of subset	
M1	254	381	215.9	342.9	29.85	149.4	71.33	353.01	635	326.12	0.55	0.66	30.4	training	Hsu (1968b)
M2	254	381	215.9	342.9	30.54	104.9	71.33	357.15	762	328.88	0.79	0.79	40.6	training	
M3	254	381	215.9	342.9	26.75	139.7	126.7	326.12	889	321.99	1.05	0.92	43.8	training	
M4	254	381	215.9	342.9	26.54	104.9	126.7	326.81	1016	318.54	1.39	1.05	49.6	training	
M5	254	381	215.9	342.9	27.99	82.6	126.7	330.95	1143	335.09	1.77	1.18	55.7	training	
M6	254	381	215.9	342.9	29.37	69.9	126.7	340.6	2288	317.85	2.09	2.36	60.1	training	
I2	254	381	215.9	342.9	45.23	98.6	71.33	348.87	635	325.43	0.84	0.66	36.0	training	
I3	254	381	215.9	342.9	44.75	127	126.7	333.71	762	343.36	1.15	0.79	45.6	training	
I4	254	381	215.9	342.9	44.95	92.2	126.7	326.12	889	315.09	1.59	0.92	58.1	training	
I5	254	381	215.9	342.9	45.02	69.9	126.7	325.43	1016	310.26	2.09	1.05	70.7	training	
I6	254	381	215.9	342.9	45.78	57.2	126.7	328.88	1143	325.43	2.56	1.18	76.7	test	
G1	254	508	215.9	469.9	29.79	187.5	71.33	339.22	508	321.99	0.4	0.39	26.8	training	
G2	254	508	215.9	469.9	30.89	120.7	71.33	333.71	635	322.67	0.63	0.49	40.3	training	
G3	254	508	215.9	469.9	26.82	155.7	126.7	327.5	762	338.53	0.87	0.59	49.6	verification	
G4	254	508	215.9	469.9	28.27	114.3	126.7	341.98	889	325.43	1.18	0.69	64.9	training	
G5	254	508	215.9	469.9	26.89	85.9	126.7	327.5	1016	330.95	1.57	0.79	72.0	training	
G6	254	508	215.9	469.9	29.92	127	126.7	349.56	1144	334.4	1.06	0.89	39.1	verification	
G7	254	508	215.9	469.9	30.96	146.1	126.7	322.67	1430	319.23	0.92	1.11	52.7	training	
G8	254	508	215.9	469.9	28.34	104.9	126.7	328.88	1716	321.99	1.28	1.33	63.3	training	
C1	254	508	215.9	215.9	27.03	215.9	71.33	341.29	381	341.29	0.22	0.3	11.3	training	
CC2	254	508	215.9	215.9	26.54	117.6	71.33	344.74	508	334.4	0.41	0.39	15.3	training	
C3	254	508	215.9	215.9	26.89	139.7	126.7	329.57	635	330.95	0.61	0.49	20.0	training	
C4	254	508	215.9	215.9	27.17	98.6	126.7	327.5	762	336.46	0.86	0.59	25.3	verification	
C5	254	508	215.9	215.9	27.23	73.2	126.7	328.88	889	328.19	1.16	0.69	29.7	verification	
C6	254	508	215.9	215.9	27.58	54.1	126.7	327.5	1016	315.78	1.57	0.79	34.2	training	

train, verify and test cases. And the cases are divided in the proportions 3:1:1 between the three subsets. In other words, among the collected data, 46, 15, and 15 are sampled randomly as training, verification, and test examples, respectively.

It is well known that neural network training algorithms are iterative, training over a period of time, and need to be repeated a number of times until a satisfactory solution is found. What is more, some difficult decisions have to be made. Among the decisions that the neural network designer must make, which of the available variables to use as inputs to the neural network is one of the most difficult. Fortunately, neural networks can learn functional relationships from examples without prior knowledge of the underlying mathematical model. In the study, therefore, a RBFN using all available variables as the input variables was developed. Nevertheless, a neural network with less input is usually preferable. Accordingly, several RBFN models that use fewer parameters as the input variables were also developed. Practically, a sensitivity analysis was carried out on the impact of each input on the neural network performance. The sensitivity analysis ranks parameters in order of importance. Sensitivity is reported separately for training and verification subsets, and the consistency of the sensitivity ratings across the two subsets is a good initial cross check on the reliability of the sensitivity analysis. The possible redundancies and interdependencies between variables imply that the sensitivity figures must be interpreted with caution since they may be quite different on another network applied to the same data set. However, it was found that certain variables consistently have high or low sensitivity, and then one can begin to identify key and unnecessary variables. Furthermore, the selection of input variables for network models was also guided by examining those variables given in the literature. Finally, five most important parameters (i.e. short dimension of closed stirrup, long dimension of closed stirrup, concrete compressive strength, steel ratio of stirrups, and steel ratio of longitudinal reinforcement) were selected to use as inputs to the neural network.

On the other hand, no specific guidelines exist on how to choose the number of neurons in the hidden layer. Therefore, the number of neurons in the hidden layer is determined through a trial-and-error process, as is normally done. After a number of trials by using different hidden neurons, several models are considered for predicting the ultimate torsional strength T_u. The architecture of

Table 2 Ranges of parameters in database

Parameters	Minimum	Maximum
x = short dimension of the cross section (mm)	160	350
y = long dimension of the cross section (mm)	275	508
x_1 = short dimension of the closed stirrup (mm)	130	300
y_1 = long dimension of the closed stirrup (mm)	216	469
f'_c = concrete compressive strength (MPa)	26	110
s = spacing of stirrups (mm)	50	215
A_t = cross-sectional area of one leg of closed stirrup (mm^2)	71	127
f_{yv} = yield strength of closed stirrups (MPa)	319	672
A_l = total area of longitudinal torsional reinforcement (mm^2)	381	3438
f_{yl} = yield strength of longitudinal torsional reinforcement (MPa)	310	638
$\rho_t = (A_t p_h)/(xys)$ (%)	0.22	2.56
$\rho_l = A_l/(xy)$ (%)	0.30	3.51
T_u = ultimate torsional strength (N-m)	11	239

Table 3 Architecture of RBFN models

Model	Inputs variables	Output variable	Number of neurons		
			Input layer	Hidden layer	Output layer
RBF-12-46-1	x, x_1, y, y_1, f'_c, s, A_t, f_{yv}, A_l, f_{yl}, ρ_t, ρ_l	T_u	12	46	1
RBF-5-28-1	x_1, y_1, f'_c, ρ_t, ρ_l	T_u	5	28	1
RBF-5-13-1	x_1, y_1, f'_c, ρ_t, ρ_l	T_u	5	13	1
RBF-5-12-1	x_1, y_1, f'_c, ρ_t, ρ_l	T_u	5	12	1
RBF-5-10-1	x_1, y_1, f'_c, ρ_t, ρ_l	T_u	5	10	1
RBF-5-9-1	x_1, y_1, f'_c, ρ_t, ρ_l	T_u	5	9	1
RBF-5-8-1	x_1, y_1, f'_c, ρ_t, ρ_l	T_u	5	8	1

the developed RBFN models is shown in Table 3. The first column in Table 3 denotes the neural network structure. For example, RBF-12-46-1 stands for the network model using RBFN with 3 layers, 12 input neurons, 1 hidden layer (with 46 hidden neurons), and 1 output neuron.

4.2. Network topology and training algorithm

As preivously mentioned, training of RBFNs takes place in distinct stages. First, the centers and width factors of the radial basis functions must be set; then the linear output layer is optimized. Consequently, the centers stored in the radial hidden layer are optimized first using a kind of unsupervised training technique (i.e. the *K*-means method). A set of data points is randomly selected from the training data set. These data points are the seeds of an incremental *K*-means clustering procedure and these *K*-means centers are used as centers in the RBFN. Then the width of the data is reflected in the radial deviations, and deviations are assigned by the *K*-nearest neighbor method. In other words, each basis's deviation is individually set to the mean distance to its *K* nearest neighbors. Hence, deviations are smaller in tightly packed areas of space, preserving detail, and higher in sparse areas of space. Once centers and deviations have been set, the linear output layer is optimized using the pseudo-inverse technique, as this is quick, and guaranteed to minimize the error if the deviations are not too small.

The data set is divided into three subsets; training, verification and test cases. To reiterate, the neural networks are trained using the training subset only. The verification subset is used to keep an independent check on the performance of the networks during training, with deterioration in the verification errors indicating over-learning. If over-learning occurs, the software stops training the network, and restores it to the state with minimum verification error. The verification error is also used by the software to select between the available networks. However, if a large number of networks are tested, a random sampling effect can kick in, and one may get a network with a good verification error which is not actually indicative of good generalization capabilities. Therefore, a third subset (i.e. the test subset) is maintained, and one can visually inspect performance after training.

4.3. Validation of RBFN models

During the training process, all RFB network models use the same training, verification, and test subsets. Basically, the performance of the developed RBFNs is measured in two aspects: one is the

root-mean-square error (RMSE) value, and the other is the coefficient of determination (R^2). On the other hand, the coefficient of determination (R^2) can be used as an index of how well the independent variables considered account for the measured dependent variable and thus testing the accuracy of the developed RFBNs.

In principle, the lower the RMSE value or the higher the R^2 value is, the better the prediction relationship will be. Judging from this, it can be seen from Table 4 that the developed RBFN models (i.e. RBF-5-28-1 and RBF-12-46-1) performance well in terms of the RMSE and R^2 values. Especially, the RMSE values of the verification and test subsets are reasonably close together, and thus the network is likely to generalize well. Besides, the R^2 values are all greater than 0.98 for the verification and test subsets. These demonstrate a near correlation between the independent variables and the measured dependent variable. In other words, the results indicate the ultimate torsional strength T_u of RC beams under pure torsion can be fairly accurately estimated using RBFNs.

The results shown in Table 4 also indicate the ultimate torsional strength T_u of RC beams under pure torsion can be fairly accurately estimated using only five input variables x_1, y_1, f'_c, ρ_t and ρ_l (i.e. short dimension of closed stirrup, long dimension of closed stirrup, concrete compressive strength, steel ratio of stirrups, and steel ratio of longitudinal reinforcement). This is quite consistent with most existing analytical methods. On the whole, however, the performance of the developed RBFN models deteriorates with the decrease of neurons in the hidden layer. Besides, inclusion of x, y, s, A_t, f_{yv}, A_l, and f_{yl} (i.e. short dimension of cross section, long dimension of cross section, spacing of stirrups, cross-sectional area of one leg of closed stirrup, yield strength of closed stirrups total area of longitudinal torsional reinforcement, and yield strength of longitudinal torsional reinforcement), in addition to x_1, y_1, f'_c, ρ_t and ρ_l, has positive effect upon the accuracy of predictions for T_u. Nevertheless, networks with fewer inputs and neurons in the hidden layer are desirable, if not too much performance is sacrificed in comparison with larger networks, as this means that less information has to be collected to use the network. Moreover, a network with fewer inputs and neurons in the hidden layer is also, in most circumstances, likely to generalize better.

4.4. Comparison with ACI design code equations

To compare the neural network results with aforementioned ACI 318 Code equations, the same training, verification and test data are used to calculate the predicted ultimate torsional strength T_{up}. Regarding all 76 specimens, the measured ultimate torsional strength (T_{ue} collected from the literature) is plotted against the predicted values, as shown in Fig. 4. To show the overall trend of correlation, the theoretical line with $T_{ue} / T_{up} = 1$ are drawn on the graphs along with the data points plotted. The nearer the points gather around the diagonal line, the better are the predicted values. Fig. 4 clearly shows that the less scatter of data around the diagonal line confirms the fact that neural network-based models are an excellent predictor for the value of T_u. While the correlation between the values of T_{ue} and T_{up}, which were obtained from Eq. (1) and Eq. (2), is more scattered. Histograms of the measured-to-predicted ultimate torsional strength ratios for training, verification, and test data are given in Fig. 5. For instance, the ranges of strength ratios for the beams examined were 0.322 to 1.381 and 0.646 to 1.144 for the ACI-05 model and RBF-12-46-1 model, respectively. For comparison purpose, the values of RMSE and R^2 of the training, verification and test results for all prediction models are also listed in Table 4. Overall, it is seen that the RBF-12-46-1 model gives the smallest RMSE and the largest R^2. In addition, all prediction models have

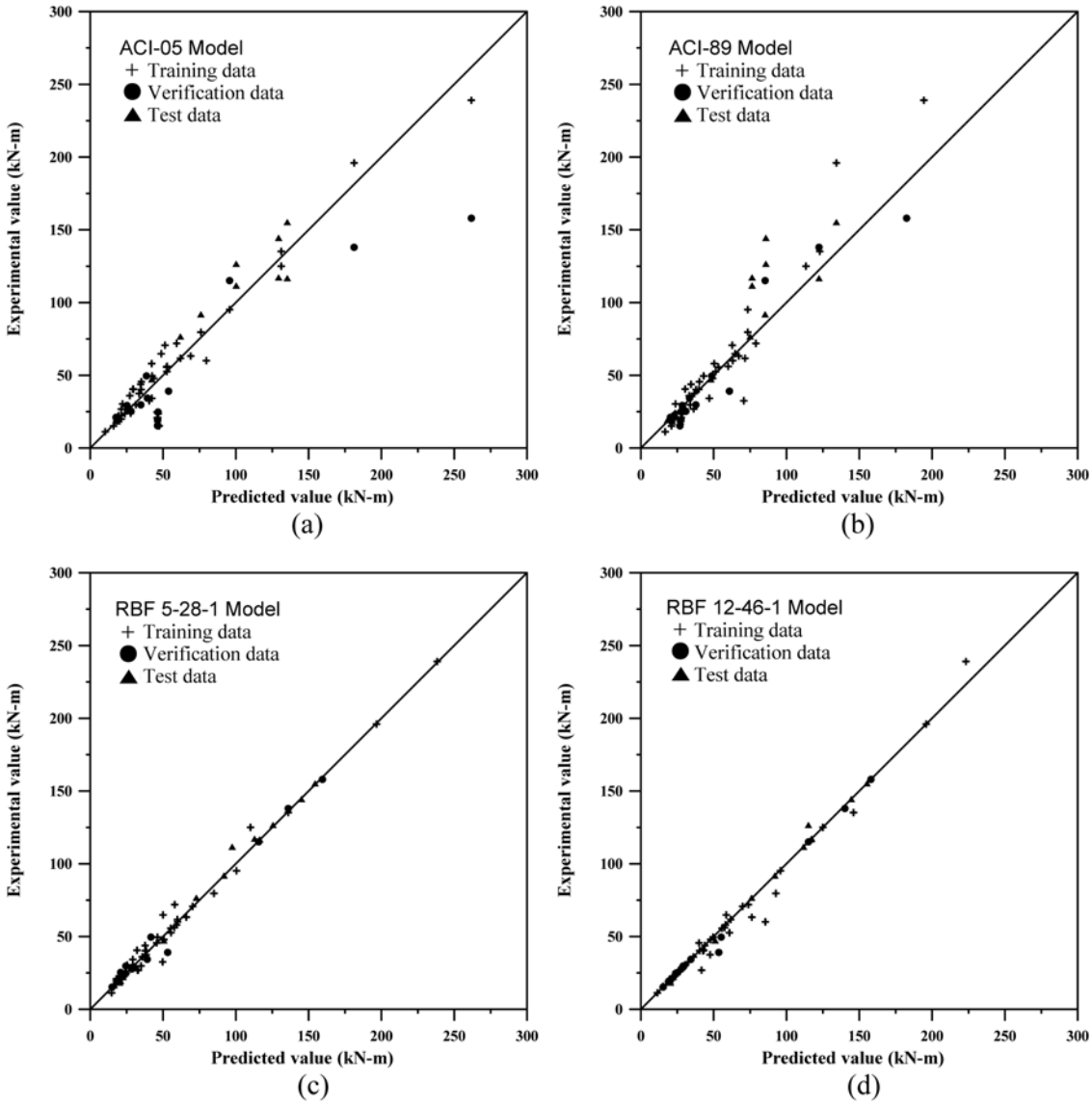


Fig. 4 Measured-versus-predicted ultimate torsional strength of RC

been compared by means of the average value (AVG), standard deviation (STD), and coefficient of variation (COV) of the strength ratios of T_{ue} / T_{up} (as shown in Table 5). The overall predictions from the RBFN models (i.e. RBF-5-28-1, RBF-5-13-1, RBF-5-12-1, RBF-5-10-1, and RBF-12-46-1) were found to be better than the ACI 318 equations. This indicates that the neural network performs better than the other methods selected in this study.

Since the ACI 318 code has been the most widespread code of practice in the design process of concrete structures, the strength ratios of T_{ue} / T_{up} from the RBF-12-46-1 model were compared with those derived from Eq. (2) for various values of $f_c^{'}$, ρ_t, and ρ_l. From Fig. 6-8, it can be observed that a large variation in the accuracy is noticed in the ACI 318-05 torsion strength prediction. For

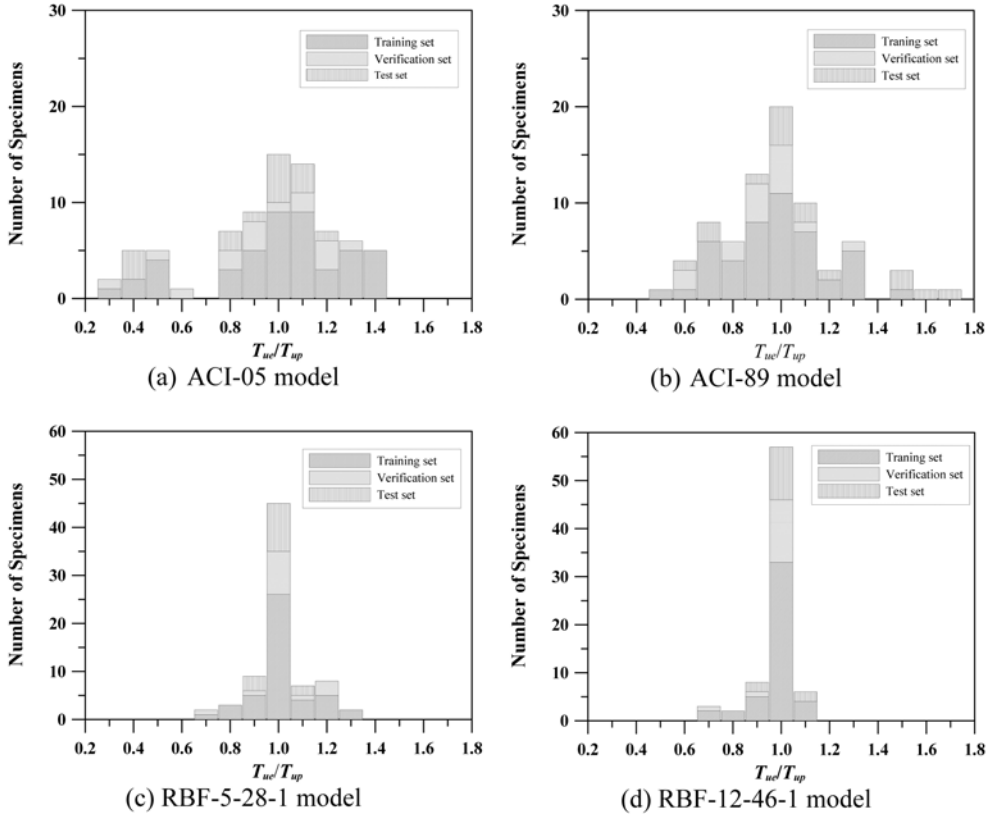


Fig. 5 Histograms of measured-to-predicted ultimate torsional strength ratios

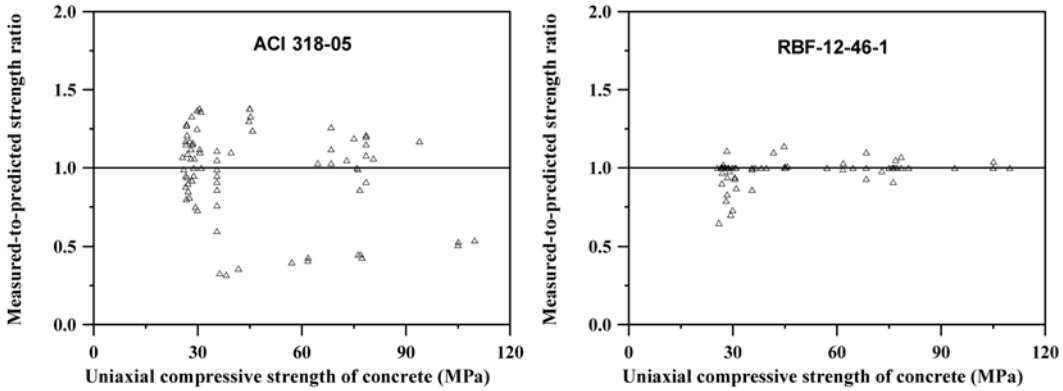
Table 4 Summary of values of RMSE and R^2

Model	Root-Mean-Square error (RMSE): kN-m			Coefficient of determination (R^2)		
	Training set	Verification set	Test set	Training set	Verification set	Test set
ACI 318-89	14.34	12.79	24.23	0.96791	0.96025	0.93034
ACI 318-05	13.12	31.49	17.93	0.95738	0.94005	0.93983
RBF-12-46-1	6.29	4.02	3.23	0.98213	0.98772	0.98836
RBF-5-28-1	5.37	4.78	4.12	0.99276	0.99447	0.99726
RBF-5-13-1	6.46	7.13	10.18	0.98939	0.98825	0.98179
RBF-5-12-1	6.47	7.17	10.18	0.98937	0.98789	0.98162
RBF-5-10-1	7.01	7.62	10.74	0.98749	0.98592	0.97925
RBF-5-9-1	7.01	8.24	9.46	0.98749	0.98825	0.98371
RBF-5-8-1	7.16	8.46	9.59	0.98695	0.98774	0.98327

example, reviewing the test specimens with ρ_t of 1.49%, as shown in Table 1 (Rasmussen and Baker 1995), it can be found that excluding the concrete strength, the cross-sectional dimensions, and strength and dimensions of the reinforcement, were constant for all beams. In other words, of the parameters that influence torsional capacity, concrete strength was the only one varied. The

Table 5 Summary of values of AVG, STD and COV for all prediction models

Model	AVG			STD			COV		
	Training set	Verification set	Test set	Training set	Verification set	Test set	Training set	Verification set	Test set
ACI 318-89	0.9681	0.9301	1.0854	0.2076	0.1905	0.3237	0.2145	0.2048	0.2982
ACI 318-05	0.9951	0.9059	0.9620	0.2866	0.2756	0.3133	0.2881	0.3042	0.3257
RBF-12-46-1	0.9746	0.9786	1.0031	0.0880	0.0745	0.0498	0.0903	0.0762	0.0496
RBF-5-28-1	1.0115	1.0237	1.0008	0.1208	0.1240	0.0551	0.1194	0.1211	0.0551
RBF-5-13-1	1.0057	1.0293	1.0557	0.1657	0.1696	0.1121	0.1648	0.1648	0.1062
RBF-5-12-1	1.0049	1.0387	1.0473	0.1704	0.1707	0.1177	0.1696	0.1643	0.1123
RBF-5-10-1	1.0025	1.0511	1.0283	0.1756	0.2241	0.1590	0.1751	0.2132	0.1546
RBF-5-9-1	1.0099	1.1110	1.1438	0.1736	0.2601	0.4651	0.1719	0.2341	0.4066
RBF-5-8-1	1.0126	1.1192	1.1423	0.1858	0.2762	0.4750	0.1835	0.2468	0.4158



(a) ACI-05 model (b) RBF-12-46-1 model

Fig. 6 Measured-to-predicted ultimate torsional strength ratios versus compressive strength of concrete

concrete strength varied between 36 and 110 MPa. The test series has shown the ultimate torsional strength of RC beams increases with the increase of concrete strength. However, according to the current torsion provision of ACI 318-05, the ultimate torsional strength of RC beams is assumed to be resisted by the closed stirrups and longitudinal steel while the torsion moment resisted by the concrete compression struts is assumed as zero. That is to say, the ACI 318-05 method does not include the effect of the concrete strength. Therefore, from Fig. 8(a) for 1.49% value of the steel ratio stirrups, it is clear that the measured-to-predicted ratios in the case of ACI-05 method yielded unconservative and scattered results. This reveals that a large variation in the accuracy is noticed in the ACI 318-05 torsion strength prediction. By contrast, the result obtained from the RBFN is the consistent one, having values close to 1 for a wide variation of those parameters.

A lot of parametric studies concerning the effects of numerous variables such as the amount of transverse and longitudinal reinforcement, the concrete strength, and the aspect ratio on the torsional behavior of RC beams have been reported in the literature. In fact, one advantage of neural network models is that parametric studies can be easily done by simply varying one input parameter and all other input parameters are set to constant values. For example, the RBF-12-46-1 method is

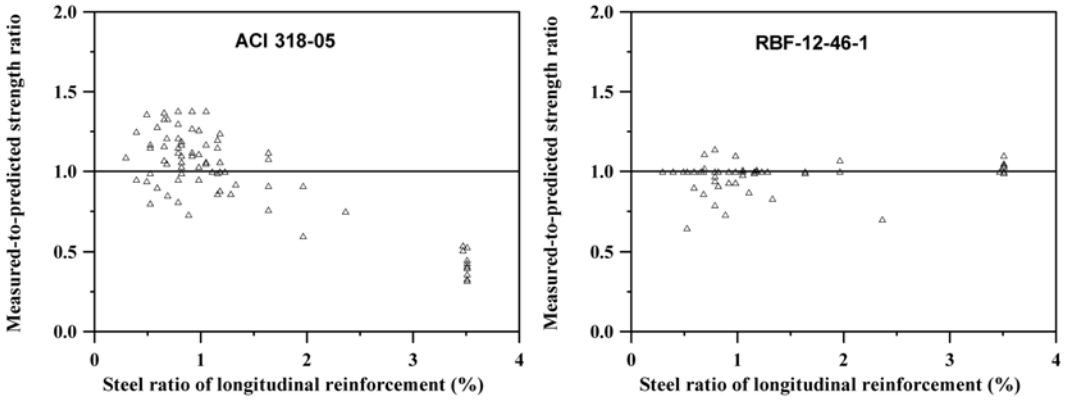


Fig. 7 Measured-to-predicted ultimate torsional strength ratios versus steel ratio of longitudinal reinforcement

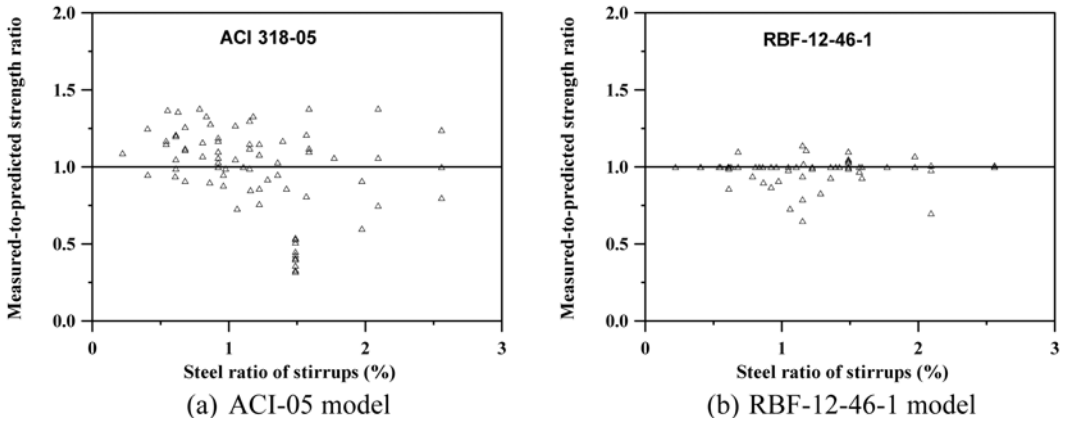


Fig. 8 Measured-to-predicted ultimate torsional strength ratios versus steel ratio of stirrups

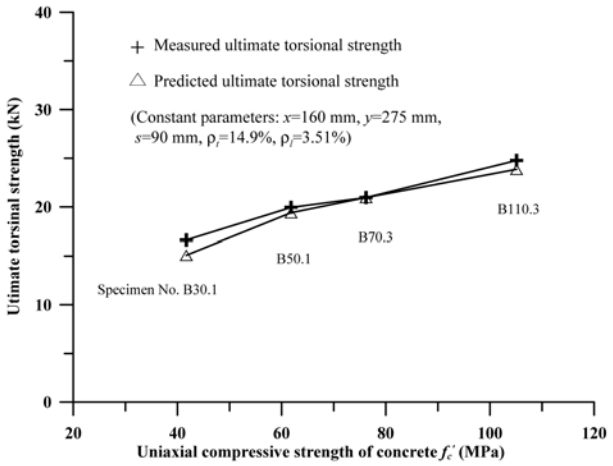


Fig. 9 Variation of ultimate torsional strength with concrete strength

considered to further investigate the effect of f_c' on the predictions of the torsional strengths of the RC beams. Fig. 9 shows an almost linear relation between the torsional strength and f_c' with the other parameters constant. Therefore, it should be noted that if the RBFN model will be used in design, then an acceptable factor of safety may be applied to the predicted value. However, the value of the factor of safety has still to be evaluated since the RBFN model can be further improved by considering more experimental data especially at high strength concrete and in the region of larger steel ratios for longitudinal reinforcement and stirrups.

5. Conclusions

The use of RBFNs for predicting the torsional strength of reinforced concrete beams is presented. The RBFNs approach can be used to predict results reasonably well and show to be an easy and sufficiently accurate method of analysis. Based on the analytical results, the following conclusions can be drawn for the present study:

1. The results clearly demonstrate that the use of RBFNs is a feasible method in predicting the ultimate torsional strength of RC beams with rectangular section subjected to pure torsion by the excellent correlation between experimental and calculated values;
2. Compared with ACI 318 equations, the RBFNs approach provides better results both in terms of root-mean-square error and coefficients of determination; and
3. The measured-to-predicted ultimate torsional strength ratio from the ACI Code was affected with the material properties of the beam. By contrast, the RBFNs keep consistent accuracy in all ranges of these variations.

Acknowledgements

The author would like to express his appreciation of the financial support received from the National Science Council, R.O.C., under Grant NSC-92-2211-E-230-004.

Notation

A_0 = gross area enclosed by shear flow path
A_l = total area of longitudinal torsional reinforcement
A_t = cross-sectional area of one leg of closed stirrup
$\mathbf{f}^n$ = target output vector
$\hat{f}_{k,}$ = estimated output
f_c' = concrete compressive strength
$(F)_{nk}$ = kth component of the nth target vector $\mathbf{f}^n$
f_{yl} = yield strength of longitudinal torsional reinforcement
f_{yv} = yield strength of closed stirrups
$h_j(\mathbf{x}^n)$= output from the jth hidden neuron for the input vector $\mathbf{x}^n$
h_0 = an extra RBF with activation value fixed at 1
$(H)_{nj}$ = outcome of the jth basis function with the nth feature vector $\mathbf{x}^n$ as input
p_h = perimeter of centerline of outmost closed transverse torsional reinforcement

s = spacing of stirrups
T_c = torsional resistance provided by concrete
T_n = nominal torsional strength (kN-m)
T_s = torsional resistance provided by steel reinforcement
T_u = ultimate torsional strength
T_{ue} = measured ultimate torsion strength
T_{up} = predicted ultimate torsional strength
W = matrix of the output layer weights
w_{k0} = a bias term at the kth output neuron
w_{kj} = weight connecting the jth basis function and the kth output
x = short dimension of the cross section
x_1 = short dimension of the closed stirrup
$\mathbf{x}^n$ = input feature vector
y = long dimension of the cross section
y_1 = long dimension of the closed stirrup
σ_j = width factor (or standard deviation) of the jth basis function
Δw_{kj} = change of weight
$\boldsymbol{\mu}_j$ = a center vector
θ = angle of compression diagonals
ρ_t = steel ratio of stirrups
ρ_l = steel ratio of longitudinal reinforcement
η = learning rate

References

ACI Committee 318 (1989), *Building Code Requirements for Structural Concrete* (ACI 318-89), *American Concrete Intitute*, Detroi.

ACI Committee 318 (2005), "Building code requirements for structural concrete (ACI 318-05) and commentary (318R-05)", *American Concrete Intitute*, Farmington Hills, Mich.

Anderson, P. (1935), "Experiments with concrete in torsion", *Transactions ASCE*, 60, 641-652.

Bishop, C.M. (1995), *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press.

Collion, C.D., Walsh, P.F., Archer, F.E., and Hall, A.S. (1965), "Reinforced concrete beams subjected to combined torsion and shear", *UNICIV Report*, No. R-14, University of New South Wales.

Consolazio, G.R. (2000), "Iterative equation solver for bridge analysis using neural networks", *Comput. Aided Civil Infrastruct. Eng.*, **15**(2), 107-119.

Cowan, H.J. (1950), "Elastic theory for torsional strength of rectangular reinforced concrete beams", *Mag. Concrete Res.,* **2**(4), 3-8.

Elfegren, L., Karlsson, I., and Losberg, A. (1974), "Torsion bending-shear interaction for concrete beams", *J. Struct. Div.,* ASCE, **100**(ST8), 1657-1676.

Fang, I.K. and Shiau, J.K. (2004), "Torsional behavior of normal- and high-strength concrete beams", *ACI Struct. J.*, **101**(3), 304-313.

Ghaboussi, J., Garrett, J.H., and Wu, X. (1991), "Knowledge-based modeling of material behavior with neural networks", *J. Eng. Mech.*, ASCE, **117**(1), 129-134.

Goode, C.D. and Helmy, M.A. (1968), "Ultimate strength of reinforced concrete beams in bending and torsion", *Torsion of Structural Concrete*, SP-18, American Concrete Institute, Detroit, 357-377.

Hajela, P. and Berke, L. (1991), "Neurobiological computational models in structural analysis and design", *Comput. Struct*., **41**(4), 657-667.

Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, Englewoods Cliffs, NJ: Macmillan.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, New Jersey.

Hopfield, J.J. (1982), "Neural network and physical systems with emergent collective computational abilities",

Proceeding of the National Academy of Science, **79**, 2554-2558.
Hsu, T.T.C. (1968a), "Torsion of structural concrete-plain concrete rectangular sections", *Torsion of Structural Concrete*, SP-18, American Concrete Institute, Detroit, 203-238.
Hsu, T.T.C. (1968b), "Torsion of structural concrete-behavior of reinforced concrete rectangular members", *Torsion of Structural Concrete*, SP-18, American Concrete Institute, Detroit, 261-306.
Hsu, T.T.C. and Mo, Y. L. (1985b), "Softening of concrete in torsional members-design and recommendations", *ACI Struct. J.*, **82**(4), 443-452.
Hsu, T.T.C. and Mo, Y.L. (1985a), "Softening of concrete in torsional members-theory and tests", *ACI Struct. J.*, **82**(3), 290-303.
Jang, J.S.R., Sun, C.T., and Mizutani, E. (1997), *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, New Jersey.
Kohonen, T. (1988), "An introduction to neural computing", *Neural Networks*, **1**, 3-16.
Koutchoukali, N.E. and Belarbi, A. (2001), "Torsion of high-strength reinforced concrete beams and minimum reinforcement requirement", *ACI Struct. J.*, **98**(4), 462-469.
Lessig, N.N (1958), "Theoretical and experimental investigation of reinforced concrete elements subjected to combined bending and torsion", *Theory of Design and Construction of Reinforced Concrete Structures*, Moscow, 73-84.
MacGregor, J.G. and Ghoneim, M.G. (1995), "Design for torsion", *ACI Struct. J.*, **92**(2), 211-218.
Moody, J.E. and Darken, C.J. (1989), "Fast learning in networks of locally tuned processing units", *Neural Computation*, **1**, 281-303.
Rasmussen, L.J. and Baker, G. (1995), "Torsion in reinforced normal and high-strength concrete beams part 1: experimental test series", *ACI Struct, J.*, **92**(1), 56-62.
Rausch, E. (1929), "Design of reinforced concrete in torsion", *Technische Hochschule*, Berlin, 53. (in German)
Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), "Learning internal representation by error propagation", *Parallel Distributed Processing*, 1, Rumelhart, D.E., and McClelland, J.L. (eds.), M.I.T. Press, Cambridge, MA, 318.
Sanad, A. and Saka, M.P. (2001), "Prediction of ultimate shear strength of reinforced-concrete deep beams using neural networks", *J. Struct. Eng.*, **127**(7), 818-28.
Schwenker, F., Kestler, H.A., Palm, G. and Höher, M. (1994), "Similarities of LVQ and RBF learning-a survey of learning rules and the application to the classification of signals from high-resolution electrocardiography", *Proc. Int. Conf. Syst*., Man Cybern, 646-651.
Tang, C.W., Chen, H.J., and Yen, T. (2003), "Modeling the confinement efficiency of reinforced concrete columns with rectilinear transverse steel using artificial neural networks", *J. Struct. Eng.*, ASCE, **129**(6), 775-783.
Yeh, I.C. (1999), "Design of high-performance concrete mixture using neural networks and nonlinear programing", *J. Comput. Civil Eng.,* ASCE, **13**(1), 36-42.
Yudin, V.K. (1962), "Determination of load-carrying capacity of rectangular reinforced concrete elements subjected to combined torsion and bending", *Institute Betona i Zhelezobeton*, Moscow, No. 6, 209-269. (in Russian)
Zhao, Z. and Ren, L. (2002). "Failure criterion of concrete under triaxial stresses using neural networks". *Comput. Aided Civil Infrastruct. Eng.,* **17**(1), 68-73.

CC