APPLICATION OF NEURAL NETWORKS FOR MODELING SHEAR STRENGTH OF REINFORCED CONCRETE BEAMS

Chao-Wei Tang¹  Tsong Yen²

¹Department of Civil Engineering,
Cheng-Shiu Institute of Technology,
Kaohsiung County 833, Taiwan, R.O.C.

²Department of Civil Engineering,
National Chung-Hsing University,
Taichung, Taiwan 402, R.O.C.

Key Words:  reinforced concrete beam, shear strength, artificial neural network.

ABSTRACT

Reinforced concrete is a nonhomogeneous, nonisotropic, and highly nonlinear material, so the real distribution of shear stresses over its cross section is a complicated problem that makes uneasy in mathematical modeling. Based on the artificial neural networks technology, this paper presents a nontraditional approach to the prediction of the ultimate shear strength of RC beams with web reinforcement. A standard back-propagation neural network (BPNN) and a multilayer-functional-link neural network (MFLNN) are used for training and testing the experimental data. A comparison study between the neural network model and five parametric models is also carried out. It was found that both the BPNN and the MFLNN are able to generalize the functional relationship between the independent variables and the measured dependent variables, and the MFLNN has better accuracy and efficiency than the BPNN. Moreover, compared with parametric models, the neural network approach provides better results. The results show that neural networks have strong potential as a feasible tool for predicting the ultimate shear strength of RC beams with web reinforcement.

應用類神經網路預測鋼筋混凝土樑之剪力強度

湯兆緯¹  顏聰²

關鍵詞：鋼筋混凝土樑，剪力強度，類神經網路。

摘 要

鋼筋混凝土是一種非均質、非等向性與非線性之材料，於承受荷重狀況下，
1. INTRODUCTION

In a reinforced concrete (RC) beam, flexure and shear combine to create a biaxial state of stress. Once the so generated principal tensile stresses exceed the tensile strength of concrete, diagonal cracks develop in the beam, eventually causing failure if the beam has insufficient shear reinforcement. Therefore, a number of theoretical and experimental studies have been conducted to analyze the flexure-shear interaction of RC beams without web reinforcement. As for RC beams with stirrups, most concrete codes (e.g., ACI 318-99 [1]; NZS 3101-1995 [2]) simply prescribe the superposition of the shear strength in absence of shear reinforcement and the additional capacity due to shear reinforcement alone. This simplification is purely for convenience to treat each as a separate calculation. In fact, concrete shear resisting mechanisms, made of the so-called “beam and arch action” contributions [3-4], can interact with stirrups in different ways [5].

The characteristic parameters for the flexure-shear interaction have been suggested over the last 50 years. These parameters include concrete compressive strength, $f_c$; shear span to effective depth ratio, $a/d$; longitudinal tension steel ratio, $\rho$; transverse steel ratio, $\rho_t$; size effect as well as the shape of beam, and loading and support conditions. On the basis of these studies, the strength prediction equation of RC members can be expressed as a function of those parameters. A multivariable nonlinear regression analysis might be performed so that the parameters are calibrated to fit the experimental results and to derive relationships among the involved parameters. Notwithstanding this, it is difficult to apply the statistical approach in a complex nonlinear system because choosing a suitable regression equation involves technique and experience. Moreover, the failure modes, the resisting mechanisms at cracked stages, and the role of various parameters presently are still under discussion and far from being settled.

By contrast, artificial neural networks are a computational tool that attempts to simulate the architecture and internal operational features of the human brain and neurons system [6-8]. Much of the success of neural networks is due to such characteristics as nonlinear processing, parallel processing, etc. Therefore, artificial neural networks have attracted considerable attention and shown promise for modeling complex nonlinear relationships. In fact, neural network modeling techniques have been rapidly applied in engineering, business, psychology and semiotics in recent years. The first journal article on civil/structural engineering applications of neural networks was published by Adeli and Yeh [9]. Since then, neural network modeling techniques have been rapidly applied in civil/structural engineering such as structural analysis and design [10-12], structural damage assessment [13-14], structural dynamics and control [15-16], seismic liquefaction prediction [17], constitutive modeling [18-20], pavement condition-rating modeling [21], and evaluating CPT calibration chamber test data [22-23].

It has been proven that the computing abilities of a
Application of Neural Networks for Modeling Shear Strength of Reinforced Concrete Beams

If the contribution of the dowel force toward the flexural resistance is ignored, the moment of resistance simplifies to [4]:

\[ M = Tjd \] (1)

where \( T \) is the tensile force in the longitudinal steel; \( j \) is the ratio between internal lever arm and effective depth; and \( d \) is the effective depth of the beam. By combining Eq. (1) with the well-known relationship between shear and the rate of change of bending moment along a beam, the following modes of internal shear resistance result:

\[ V = \frac{dM}{dx} = \frac{d}{dx} (Tjd) = jd \frac{dT}{dx} + Td \frac{dj}{dx} \] (2)

The term \( jd\frac{dT}{dx} \), called “beam action”, is the shear resulting from a gradient in steel tensile force on constant lever arm; while the term \( Td\frac{dj}{dx} \), called “arch action”, is the shear resulting from a constant steel tensile force acting on a varying lever arm. By starting from the sum in Eq. (2), Bazant and Kim [30] have developed a mechanic analysis for each of the two modes of shear resistance. On the basis of an extensive analysis on 18 experimental investigations and 297 beams, they provided for concrete ultimate shear strength \( V_c \) the sum of beam action and arch action contributions as

\[ V_c = V_b + V_a \] (3)

where

If the contribution of the dowel force toward the compression zone, \( V_{C2} \),

- Interlocking action of aggregates along the rough concrete surfaces on each side of the crack, \( V_{ag} \),
- Dowel action of the longitudinal reinforcement, \( V_d \).

For longitudinally reinforced beams without stirrups, after an inclined crack has formed, the shear transfer occurs by a combination of the following principal mechanisms (as shown in Fig. 1):

1. Shear resistance of uncracked concrete in the compression zone, \( V_{C2} \).
2. Interlocking action of aggregates along the rough concrete surfaces on each side of the crack, \( V_{ag} \).
3. Dowel action of the longitudinal reinforcement, \( V_d \).

![Fig. 1 Various components of the shear transfer mechanisms in beam without stirrups](image-url)
\[ V_b = 0.835 \xi^{1/3} f_c^{2/3} bd \]  
(4)

\[ V_a = 206.9 \xi^{3/4} \left( \frac{a}{d} \right)^{3/2} bd \quad (a = \text{shear span}) \]  
(5)

\[ \xi = 1/\sqrt{1+(d/25d_a)} \quad (d_a = \text{maximum aggregate size}) \]  
(6)

In which \( V_b \) is the beam action contributions to shear strength with \( f_c \) in MPa; and \( V_a \) is the arch action contributions to shear strength.

In the case of a beam with vertical stirrups, the forces acting on the portion of such a beam between the crack and the nearby support are shown in Fig. 2. They are the same as those of Fig. 1 except that each stirrup traversing the crack exerts a force \( A_s f_s \) on the given portion of the beam. Here \( A_s \) is the cross-sectional area of the stirrup, and \( f_s \) is the tension stress in the stirrup. By Mörsch’s [31] pin-jointed truss model, the resultant force \( V_s \) of all stirrups across the diagonal crack in Fig. 2 is calculated as \( n A_s f_s \), \( n \) being the number of stirrups traversing the crack. Taking into account that \( n = (jd/cot \alpha)/s \), we obtain

\[ V_s = \frac{A_s f_s jd \cot \alpha}{s} \]  
(7)

where \( \alpha \) is the angle of the diagonal compression struts to the horizontal, and \( s \) is the spacing between stirrups. Assuming that the stirrup stress \( f_s \) equals the yielding

value \( f_y, j = 1, \) and \( \alpha = 45^\circ, \) and setting \( \rho_c = A_c/bs, \) Eq. (7) provides

\[ V_s = \frac{A_s f_y d}{s} = \rho_c f_y bd \]  
(8)

In most concrete codes, the additional capacity due to shear reinforcement \( V_s \) is added to the ultimate shear strength \( V_c \) in absence of shear reinforcement. Then, at failure, the nominal ultimate shear strength \( V_n \) is given by

\[ V_n = V_c + V_s \]  
(9)

However, Russo and Puleri [5] reported that a complex interaction exists between beam, arch and truss mechanisms involved at failure, which describes that the inclusion of stirrups in a beam may cause variable effectiveness in relation to loading condition and geometrical-mechanical beam characteristics. Consequently, they provided the “stirrup effectiveness function” to take into account not only a truss mechanism variable contribution but also some enhancement of beam action contribution due to stirrup inclusion. The proposed nominal ultimate shear strength can be computed as

\[ V_n = V_c + \psi V_s \]  
(10)

where \( \psi \) is the ratio of effective shear strength with stirrup inclusion to the conventional one.
3. EXISTING DESIGN CODES AND EMPIRICAL METHODS FOR SHEAR STRENGTH

Several design codes [1-2, 31] and empirical equations in the literature are available for predicting the shear strength of RC beams with web reinforcement. Although most of them provide simple superposition of stirrup and concrete capacities, ignoring any stirrup influence on concrete mechanisms and/or any dependence of stirrup action on the failure, some prominent methods among them, which were selected and used in this study for comparison with the results from the neural network model, are outlined in the following.

3.1 ACI Code Equation

The equivalent ACI 318-99 equation [1] for the nominal ultimate shear strength in a RC beam with web reinforcement is given by

\[ V_n = \left( 0.16 \sqrt{f'_c} + 1.72 \rho_{ld} \frac{V_{ld}}{M_{ld}} \right) b d + \rho_s f_s b d \ (\text{mm-N}) \]  (11)

where \( b \) = breadth of beam; \( d \) = effective depth of beam; \( f'_c \) = cylinder compressive strength of concrete; \( f_s \) = yield strength of vertical steel; \( \rho \) = longitudinal tensile steel ratio; \( \rho_s \) = transverse steel ratio; and \( V_n, M_n \) = shear force and moment at the critical section, respectively. The first term on the right hand side of Eq. (11) is the concrete contribution, which must not be taken greater than \( 0.3 f'_c b d \) and the second term is the stirrup contribution which must not be greater than \( 0.667 f'_c b d \).

3.2 New Zealand Code Equation

The New Zealand concrete structures code [2] is applicable for \( f'_c \) up to 100 MPa, for members not resisting earthquake forces. The nominal ultimate shear strength in a RC beam with web reinforcement is given by

\[ V_n = (0.07 + 10 \rho) \sqrt{f'_c} b d + \rho_s f_s b d \ (\text{mm-N}) \]  (12)

The first term on the right hand side of Eq. (12) must be within the maximum and minimum limits of \( 0.2 \sqrt{f'_c} b d \) and \( 0.08 \sqrt{f'_c} b d \), respectively.

3.3 Zsutty Equation

Zsutty [33] recognized that the shear test data are not homogeneous due to the fact that there are two separate types of beam behavior. Accordingly, he segregated the data and performed separate regression analysis for short beams and long beam. The ultimate shear strength prediction can be computed as

\[ V_n = 2.17 \left( f'_c \frac{d}{a} \right)^{1/3} b d + \rho_s f_s b d \ (\text{mm-N}) \]  (13)

The first term on the right hand side of Eq. (13) is to be multiplied by \( 2.5(d/a) \) when \( a/d < 2.5 \).

3.4 Rebeiz Equation

Rebeiz [34] presents a shear strength prediction equation for RC members without web reinforcement. It uses the techniques of dimensional analysis, interpolation function, and multiple regression analysis. The corresponding ultimate shear strength can be computed as follow:

\[ V_n = \left[ 0.4 + (10 - 3 A_d) \sqrt{\frac{f'_c}{4}} \frac{d}{a} \right] b d + \rho_s f_s b d \ (\text{mm-N}) \]  (14)

where \( A_d = a/d \) for \( 1 < a/d < 2.5 \) or \( A_d = 2.5 \) for \( a/d > 2.5 \).

3.5 Russo and Puleri Equation

As stated previously, Russo and Puleri [5] introduced the “stirrup effectiveness function” to assess the shear strength of RC beams with stirrups. The proposed nominal ultimate shear strength can be computed as
\[ V_n = 0.833 \xi \sqrt{\frac{b}{d}} bd + \psi \sqrt{\frac{d}{25d_a}} \] (mm-N) \hspace{1cm} (15)

where

\[ \xi = 1/\sqrt{1 + \left( \frac{d}{25d_a} \right)} \]

\[ \chi = \sqrt{f_c} + 250 \sqrt{\rho \left( \frac{d}{a} \right)^3} \] \hspace{1cm} (16)

\[ \psi = \frac{1.67 \sqrt{f_c}}{\sqrt{f'_c} + 250 \sqrt{\rho (d/a)^3}} \] \hspace{1cm} (17)

4. NEURAL NETWORKS

Artificial neural networks generally consist of a number of artificial neurons (or processing units) that are arranged logically into two or more layers and interact with each other via weighted connections to constitute a network. Most neural network applications are based on the error back-propagation algorithm. Fig. 3 shows a typical back-propagation neural network with an input layer, an output layer, and one hidden layer, where \( X_i \), \( X_n \), are the \( N \) components of input vector \( X \), \( W_p \) and \( W_y \) are the connection weights between neurons of different layers, \( \Theta_i \) is the bias assigned to neuron \( j \) in the hidden layer, and \( \Theta_j \) is the bias assigned to neuron \( k \) in the output layer. Neurons in the input layer represent the possible influential factors that affect the network outputs, while the output layer contains one or more neurons that produce the network outputs. Layers between the input and output layers are called hidden layers and contain a large number of hidden neurons.

Fig. 4 shows two typical neurons selected from input layer and hidden, or output, layers of a neural network.

For the input layer, its neurons transfer the incoming signal \( X_i \) directly to the hidden layers (with no calculations happening):

\[ Y_i = X_i \] \hspace{1cm} (18)

where the subscript \( i \) denotes the \( i \)th neuron in the input layer. For the hidden or output layer, each neuron forms a weighted sum

\[ \sum_{j=1}^{N} W_{ji} X_i \] \hspace{1cm} (19)
of $N$ inputs from previous layer, and a bias is $\theta_j$ added.

$$U_j = \sum_{i=1}^{N} W_{ji} X_i + \theta_j$$

(20)

where $W_{ji}$ is termed the weighted coefficient, and the subscript $ji$ denotes that $W_{ji}$ is the connection weight on the link from neuron $i$ in the previous layer to neuron $j$ in the current layer. Then the sum becomes the input signal of the processing unit. Processing units process and pass the results through an activation function $F$ to obtain its output $Y_j$ as follows:

$$Y_j = F(U_j) = F\left(\sum_{i=1}^{N} W_{ji} X_i + \theta_j\right)$$

(21)

Many activation functions are available but it is common to use a sigmoid function [35] given by the following equation

$$F(U_j) = \frac{1}{1 + e^{-\lambda U_j}}$$

(22)

where $\lambda$ is a constant and controls the shape of the activation function. The sigmoid function has convenient property that the derivative $F'(U_j) = \lambda F(U_j)[1-F(U_j)]$ is easy to form so that little extra calculation is needed to find $F'(U_j)$.

As the name of back-propagation networks indicates, propagation takes place in a feed-forward manner from the input layer to the output layer when a set of input patterns is presented to the network, and backward error propagation begins at the output layer using a learning mechanism to adjust the weights and biases when errors propagate through the intermediate layers toward the input layer. Taking the $k$th neuron in the output layer as an example (Fig. 3), the error $E$ between the calculated values $O_k$ and the desired values $T_k$ of output layer neurons may be defined as

$$E = \frac{1}{2} \sum_{k=1}^{N} (O_k - T_k)^2$$

(23)

where

$$O_k = F(U_k) = F\left(\sum_{j=1}^{N} W_{kj} Y_j + \theta_k\right)$$

(24)

The learning mechanism of back-propagation networks is a generalized delta rule [36] that uses the
gradient-descent method to minimize the error between the actual and the desired output. Therefore, from hidden to output, the modification of weights and biases are represented respectively by the following expressions:

\[ \Delta W_{jk} = \eta \delta_k Y_j \]  
\[ \Delta \theta_j = \eta \delta_j \]

where

\[ \eta = \text{the learning rate} \]

\[ \delta_k = (T_k - O_k)F'(U_j) \]

And from input to hidden

\[ \Delta W_{ji} = \eta \delta_j X_i \]  
\[ \Delta \theta_j = \eta \delta_j \]

where

\[ \delta_j = W_{kj} \delta_k F'(U_j) \]

The training algorithm may be improved by adding momentum terms into Eq. (25) to Eq. (28), and then the weights and biases are adjusted respectively by the following expressions:

\[ W_{kj}(t+1) = W_{kj}(t) + \eta \delta_k Y_j + \beta [W_{kj}(t) - W_{kj}(t-1)] \]  
\[ \theta_k(t+1) = \theta_k(t) + \eta \delta_k + \beta [\theta_k(t) - \theta_k(t-1)] \]

\[ W_{ji}(t+1) = W_{ji}(t) + \eta \delta_j X_i + \beta [W_{ji}(t) - W_{ji}(t-1)] \]  
\[ \theta_j(t+1) = \theta_j(t) + \eta \delta_j + \beta [\theta_j(t) - \theta_j(t-1)] \]

where \( t \) denotes the learning cycle, and \( \beta \) is the momentum factor.

The process of forward and backward propagation continues until either the error is reduced to an acceptable level or the program runs for a specified time. Usually, the number of iterations is specified as an acceptable error in the Root-Mean-Square (RMSE) by Dayhoff [37]. Once the network is trained and converges, the test set is presented to the network sequentially to verify the reliability and accuracy of the network performance. Although the back-propagation learning algorithm is widely used, this algorithm has a slow rate of learning. The number of iterations for learning an example is often in the order of thousands. Moreover, the convergence rate is highly dependent on the choice of the values of learning and momentum ratios encountered in this algorithm. In order to improve the efficiency and accuracy of the back-propagation networks, many other efficient learning methods are derived [38-42]. A detailed explanation of back-propagation neural network is beyond the scope of this paper. However, the basic algorithm for back-propagation networks is described in related references [43-45].

5. NEURAL NETWORK MODELS FOR SHEAR STRENGTH

The basic strategy for developing a neural-based model for material behavior is to train a neural network on the results of a series of experiments using that material [18]. If the experimental results contain the relevant information about the materials behavior, then the trained neural network will contain sufficient information about the material behavior to qualify as a material model. Such a trained neural network not only would be able to reproduce the experimental results it was trained on, but through its generalization capability it should be able to approximate the results of other experiments. In other words, once trained, the network provides rapid mapping of a given input into the desired output quantities. Details on the establishment of neural network models for shear strength, along with sources of the data that are used in the development, are described below.

5.1 Neural Network Models and Selection of Input Parameters

In this study, a commercially available software
package [46] was used to develop artificial neural networks by which the ultimate shear strength of RC beams with stirrups can be predicted. The BPNN model, which is a standard back-propagation neural network, is used mainly for comparison purposes, whereas the MFLNN model is a modification of the standard back-propagation neural network, where enhancements with logarithm neurons and exponent neurons in the input and output layers are used to improve the network’s performance, including efficiency and accuracy.

Selection of input parameters for a network model could be guided by examining those parameters and diagrams given in the aforementioned references. After a thorough study, the following models are considered for predicting the ultimate shear strength $V_n$:

- **Nine-inputs Model**

$$V_n = f\left[\frac{a}{d}, f_c, f_y, f_y', \rho_c, f_y', \rho, \rho', b, d\right]$$  \hspace{1cm} (33)

- **Seven-inputs Model**

$$V_n = f\left[\frac{a}{d}, f_y, \sqrt{f_c}, \rho, \rho', \rho_y, f_{y'}, b, d\right]$$  \hspace{1cm} (34)

- **Five-inputs Model**

$$V_n = f\left[\frac{f_y d}{a}, \rho_c f_{y'}, \rho', b, d\right]$$  \hspace{1cm} (35)

### 5.2 Generation of Data

The process consists of initially collecting the required data. The experimental data include 108 RC beams results, which are taken from the tests carried out by Clark [24], Bresler and Scordelis [25], Bresler and Scordelis [26], Placas and Regan [27], Mattock and Wang [28], and Ozcebe et al [29]. The experimental data collected from the literature cover the geometrical and material properties, and shear strength of the specimens, which are simply supported rectangular beams subjected to two point loads or single point load acting symmetrically with respect to the centerline of the span. The ranges of these data are listed in Table 1. The data are then separated into two sets, one for training and the other for testing. Among the collected examples, 83 are sampled randomly as training examples, and the remaining 25 are regarded as testing examples.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressive strength of concrete ($f_c'$): MPa</td>
<td>12.76</td>
<td>82.00</td>
</tr>
<tr>
<td>Breadth of beam (b): mm</td>
<td>150.0</td>
<td>307.3</td>
</tr>
<tr>
<td>Effect depth of beam (d): mm</td>
<td>272.0</td>
<td>466.1</td>
</tr>
<tr>
<td>Shear span to effective depth ratio (a/d)</td>
<td>1.17</td>
<td>6.98</td>
</tr>
<tr>
<td>Longitudinal tensile steel ratio ($\rho$)</td>
<td>0.0098</td>
<td>0.0443</td>
</tr>
<tr>
<td>Longitudinal compressive steel ratio ($\rho'$)</td>
<td>0</td>
<td>0.0298</td>
</tr>
<tr>
<td>Nominal shear stress provided by stirrups ($\rho f_y'$): MPa</td>
<td>0.033</td>
<td>4.04</td>
</tr>
<tr>
<td>Yield strength of longitudinal tensile steel ($f_t'$): MPa</td>
<td>320.6</td>
<td>696.9</td>
</tr>
<tr>
<td>Yield strength of longitudinal compressive steel ($f_c'$): MPa</td>
<td>0</td>
<td>620.6</td>
</tr>
<tr>
<td>Measured ultimate shear strength ($V_{exp}$): kN</td>
<td>70.06</td>
<td>503.10</td>
</tr>
</tbody>
</table>
5.3 Training and Testing the Network

Training means to present the network with the experimental data and have it learn, or modify its weights and biases, such that it correctly reproduces the shear strength of beams. However, no specific guidelines exist on how to choose the network topology, the number of layers and the number of neurons. Therefore, training the network successfully requires many choice and training experiences. First, to avoid the slow rate of learning near the end points of the range, the input and output data were normalized. Second, the network is initialized with randomly distributed weights and biases when starting to train neural networks in the present study. The network configuration was arrived after watching the performance of different configurations for a fixed number of cycles. Then, learning parameters were changed and learning processes were repeated.

As mentioned previously, two programs (BPNN and MFLNN) have been used to model the ultimate shear strength of RC beams with web reinforcement. All programs use the same training and testing data. After a number of trials by using different hidden neurons, two models for each neural network are used. The values of network parameters considered in this approach are as follows:

- Number of Hidden Layers = 1
- Number of Training Examples = 83
- Number of Testing Examples = 25
- Learning Cycles: Depend on the model used
- Range of Weights = -0.3 to 0.3
- Random Seed = 0.456
- Learning Rate = 0.5
- Learning Rate Reduced Factor = 0.95
- Learning Rate Minimum Bound = 0.1
- Momentum Factor = 0.5
- Momentum Factor Reduced Factor = 0.95
- Momentum Factor Minimum Bound = 0.1

On the other hand, to avoid over-training, the convergence criterion adopted in this study depends on whether the RMSE value of the testing data has reached its minimum. Mathematically, the RMSE is computed as follow [37]:

$$RMSE = \sqrt{\frac{\sum_{p} \sum_{j} (T_{jp} - Y_{jp})^2}{M \cdot S}}$$

(36)

where $T_{jp}$ and $Y_{jp}$ are the desired and computed outputs for the $j$th neuron of the $p$th training instance, $M$ is the number of instances in training set, and $S$ is the number of neurons in the output layer. Besides, 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 trained network.

6. RESULTS AND DISCUSSION

6.1 Validation of Neural Network and Comparison

To provide a measure of the output network accuracy over the number of training iterations and to evaluate the confidence in the performance of the trained network, the RMSE and $R^2$ values of the testing data by the six network models are listed in Table 2. The first column in Table 2 denotes the neural network structure. For example, B5-6-1 stands for the BP neural network with 3 layers, 5 input neurons, 1 hidden layer with 6 hidden neurons, and 1 output neuron; M5-6-1 stands for the MFL neural network with 3 layers, 5 input neurons, 1 hidden layer with 6 hidden neurons, and 1 output neuron. In theory, the lower the RMSE value or the higher the $R^2$ value is the better the prediction relationship will be. Judging from this, the results in Table 2 indicate that the parameter $V_n$ can be fairly simulated using 5 and 7 input parameters. By contrast, the models with 9 input parameters significantly improve the accuracy of predictions. On the whole, all the models produced satisfactory results in terms of the coefficient of determination ($R^2$) and the RMSE value. In other words, the trained networks are shown to be able to generalize the functional relationship between the independent variables and the measured dependent variable.

Although the BPNN generally shows a good
correction between the predicted and expected results, a closer examination shows some scatter in the results for small value outputs up to approximately 80 kN, and slightly less scatter for larger value outputs. The reason is that data pairs with larger error tend to play a more dominant role in guiding the evolution of the weight matrices. By contrast, the MFLNN can reduce the difference between the smallest output value and the largest output value by using a nonlinear transformation of the output values in training data, and thus yielding better results. On the other hand, judging by the number of learning cycles, as shown in Table 2, the MFLNN is superior to the BPNN in speed of learning because the MFLNN takes fewer iterations to reach the RMSE, while the BPNN takes more iterations to reach it. Overall, it can be concluded from Table 2 that in terms of the RMSE values, the \( R^2 \) values, and the number of learning cycles, the MFLNN provides better prediction than the BPNN.

Table 2. Performance of the various neural network models in predicting \( V_n \)

<table>
<thead>
<tr>
<th>Model</th>
<th>( R^2 ) (Testing Set)</th>
<th>RMSE: kN (Testing Set)</th>
<th>Number of Learning Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>B5-6-1</td>
<td>0.9691</td>
<td>21.71</td>
<td>6730</td>
</tr>
<tr>
<td>M5-6-1</td>
<td>0.9895</td>
<td>13.66</td>
<td>3300</td>
</tr>
<tr>
<td>B7-10-1</td>
<td>0.9452</td>
<td>29.59</td>
<td>15340</td>
</tr>
<tr>
<td>M7-10-1</td>
<td>0.9523</td>
<td>29.80</td>
<td>1440</td>
</tr>
<tr>
<td>B9-12-1</td>
<td>0.9834</td>
<td>16.12</td>
<td>11100</td>
</tr>
<tr>
<td>M9-15-1</td>
<td>0.9881</td>
<td>13.54</td>
<td>3890</td>
</tr>
</tbody>
</table>

6.2 Comparison with Parametric Models

Table 2 shows that the RMSE and \( R^2 \) values of the developed Model M9-15-1 are 13.54 kN and 0.9881 for 25 beam specimens, which were reserved for testing the network. This indicates a significant correlation between the independent variables (i.e., \( \alpha, \beta, f_c, f_y, \rho, f_y, P, \rho, b, \) and \( d \)) and the measured dependent variable (i.e., the experimental ultimate shear strength \( V_{exp} \)). Consequently, the results obtained by Model M9-15-1 are selected to compare with those from the aforementioned parametric models.

To compare the M9-15-1 model results with five existing parametric models, the same training and testing data are used to calculate the predicted ultimate shear strength \( V_{pred} \). The experimental ultimate shear strength \( V_{exp} \) collected from the literature are plotted against the predicted values \( V_{pred} \) calculated by the aforementioned prediction equations, i.e., Eq. (11) to Eq. (15), and Model M9-15-1, as shown in Fig. 5 for the training data and Fig. 6 for the testing data, respectively. In principal, the nearer the points gather around the diagonal line (i.e., the theoretical line with \( V_{exp}/V_{pred} = 1) \), the better are the predicted values. It can be seen from Fig. 5 and Fig. 6 that the least scatter of data around the diagonal line confirms the fact that neural network is an excellent predictor of the ultimate shear strength, while the correlation between the experimental strength and the predicted values obtained from Eq. (11) to Eq. (15) are more scattered. In other words, the developed Model M9-15-1 produced the most accurate estimate of \( V_n \). It should be pointed out that the experimental strength is almost always higher than that predicted analytically using Eq. (11) (ACI Code equation) and Eq. (12) (New Zealand Code equation). Taking Fig. 5(a) as an example, only two out of 83 strength ratios are marginally smaller than 1. This is quite reasonable because the ACI Code approach is a design method and thus always conservative.

For comparison purpose, the RMSE and \( R^2 \) values of the training and testing results for the six prediction models are listed in Table 3. In the case of testing data, although all parametric models yielded satisfactory \( R^2 \) value, which refers to the linear regression of the experimental strength \( V_{exp} \) versus the predicted strength \( V_{pred} \), the RMSE values of the predicted strength using Eq. (11) and Eq. (12) are nearly 4 times the value of Model M9-15-1. It is seen that Model M9-15-1 gives the smallest RMSE values (15.92 kN for the training set and 13.54 kN for the testing set) and the largest \( R^2 \) values (0.9878 for the training set and 0.9881 for the testing set). In addition, the six prediction models have been
compared by means of the average value (AVG), standard deviation (STD), and coefficient of variation (COV) of the $V_{exp}/V_{pred}$ ratios, as shown in Table 4. As was expected, the results again verify that the neural network model possesses the least COV value of 6.06% (with AVG = 0.9980 and STD = 0.0604) and 9.160% (with AVG = 1.0196 and STD = 0.0934) for the training and testing sets, respectively. On the basis of these results, it can be concluded that the neural network performs better than the other methods selected in this study.

It should be noted that most equations for the prediction of shear strength are divided into two categories by the $a/d$ value: one for slender beams ($a/d \geq 2.5$) and one for short beams ($a/d < 2.5$). For short beams, the shear strength equations are usually formed by the multiplication of the equations for slender beams by a linear arch action factor to account for the increase in shear strength due to the effect of arch action. By contrast, the use of neural networks does not necessitate any such modification because the neural networks can map the gray boundaries between any two classes automatically as it computes through classification. In other words, the use of a neural network can cover the entire range of $a/d$ employed for model fitting in a single and consistent manner, whereas other methods often have two separate equations for slender and short beams.

### Table 3. Summary of the values of RMSE and $R^2$ for all the models

<table>
<thead>
<tr>
<th>Prediction Model</th>
<th>Coefficient of Determination ($R^2$)</th>
<th>Root-Mean-Square Error (RMSE): kN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training set</td>
<td>Testing set</td>
</tr>
<tr>
<td>Eq. (11): ACI Code</td>
<td>0.8028</td>
<td>0.9825</td>
</tr>
<tr>
<td>Eq. (12): New Zealand Code</td>
<td>0.7620</td>
<td>0.9750</td>
</tr>
<tr>
<td>Eq. (13): Zsutty</td>
<td>0.8956</td>
<td>0.9757</td>
</tr>
<tr>
<td>Eq. (14): Rebeiz</td>
<td>0.9269</td>
<td>0.9748</td>
</tr>
<tr>
<td>Eq. (15): Russo-Puleri</td>
<td>0.9368</td>
<td>0.9875</td>
</tr>
<tr>
<td>M9-15-1: Neural network</td>
<td>0.9878</td>
<td>0.9881</td>
</tr>
</tbody>
</table>

### Table 4. Comparison between the various shear strength prediction equations

<table>
<thead>
<tr>
<th>Prediction Model</th>
<th>Based on experimental-to-calculated ratio ($V_{exp}/V_{pred}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Value (AVG)</td>
</tr>
<tr>
<td></td>
<td>Training set</td>
</tr>
<tr>
<td>Eq. (11): ACI Code</td>
<td>1.5217</td>
</tr>
<tr>
<td>Eq. (12): New Zealand Code</td>
<td>1.5179</td>
</tr>
<tr>
<td>Eq. (13): Zsutty</td>
<td>1.1418</td>
</tr>
<tr>
<td>Eq. (14): Rebeiz</td>
<td>1.1988</td>
</tr>
<tr>
<td>Eq. (15): Russo-Puleri</td>
<td>1.1075</td>
</tr>
<tr>
<td>M9-15-1: Neural network</td>
<td>0.9980</td>
</tr>
</tbody>
</table>
Fig. 5 Experimental-versus-predicted ultimate shear strength (training data)
Fig. 6  Experimental-versus-predicted ultimate shear strength (testing data)
7. CONCLUSIONS

This study has explored the potential application of neural network techniques to predict the ultimate shear strength of RC beams with web reinforcement. It was found that both the BPNN and the MFLNN are able to generalize the functional relationship between the independent variables and the measured dependent variable, and the MFLNN has better accuracy and efficiency than the BPNN due to the feature of the MFLNN that the logarithm and exponent neurons in the input and output layers are used to improve the network’s performance. Compared with parametric models, the neural network approach provides better results. Besides, the complexity of the input-output mapping can be easily accomplished in neural network-based models by change of the architecture of the network, whereas explicit forms need to be chosen by trial and error process in traditional regression-based models. Furthermore, most equations for the prediction of shear strength are divided into two categories by the \( a/d \) value, while the use of neural networks can map the gray boundaries between slender and short beams automatically as it computes through classification.

It should be pointed out that like other data-fitting techniques, the neural network only processes predictive capacity within the range of data employed for model fitting. Consequently, it is unable to state that the network performs correctly for all ranges. However, this defect can readily be eliminated if all ranges are covered in the training data. Finally, in light of the promising results, it is believed with reasonable confidence that the application of neural network techniques to other areas of structural engineering can open new directions for further research. Particularly in the problems that have more than one existing calculation methods or the one based on only empirical approximations.

**LIST OF NOTATIONS**

- \( A_s \): cross-sectional area of the stirrup;
- \( a/d \): shear span to effective depth ratio;
- \( b \): breadth of beam;
- \( c \): depth of compression zone above the tip of the diagonal crack;
- \( d \): effective depth of beam;
- \( d_a \): maximum aggregate size;
- \( F \): sigmoid function;
- \( f' \): concrete compressive strength;
- \( f_t \): tensile strength of concrete;
- \( f' \): tension stress in the stirrup;
- \( f_y \): yield strength of longitudinal tensile steel;
- \( f_y' \): yield strength of transverse steel;
- \( h \): depth of beam;
- \( h' \): effective depth of beam;
- \( L \): effective span of beam;
- \( M \): number of instances in training set;
- \( M_a \): moment at the critical section;
- \( N \): number of component of the input vector \( X \);
- \( n \): number of stirrups traversing the crack;
- \( O_k \): calculated output of neuron \( k \);
- \( R^2 \): coefficient of determination;
- \( s \): spacing between stirrups;
- \( S \): number of processing elements in the output layer;
- \( T \): tensile force in the longitudinal steel;
- \( T_p \): desired output;
- \( T_k \): desired output of neuron \( k \);
- \( V \): support reaction;
- \( V_a \): arch action contributions to shear strength;
- \( V_{ag} \): shear contributions due to aggregate interlock;
- \( V_b \): beam action contributions to shear strength;
- \( V_c \): ultimate strength for concrete in shear;
- \( V_{Ct} \): shear contributions due to concrete;
- \( V_d \): shear contributions due to dowel action;
- \( V_{op} \): measured ultimate shear strength;
- \( V_n \): nominal ultimate shear strength in a RC
beam with web reinforcement;

\[ V_{\text{pred}} \] = predicted ultimate shear strength;
\[ V_a \] = additional capacity due to shear reinforcement;
\[ V_s \] = shear force at the critical section;
\[ W_{ji} \] = weighted coefficient between neurons of different layers;
\[ X \] = input vector;
\[ X_i \] = ith component of input vector;
\[ Y_p \] = computed output;
\[ \alpha \] = angle of the diagonal compression struts to the horizontal;
\[ \beta \] = momentum factor;
\[ \chi \] = function appearing in Eq. (16);
\[ \lambda \] = a constant;
\[ \xi \] = function appearing in Eq. (6);
\[ \psi \] = function appearing in Eq. (17);
\[ \rho \] = longitudinal tensile steel ratio;
\[ \rho' \] = longitudinal compressive steel ratio; and
\[ \rho_\gamma \] = transverse steel ratio.

REFERENCES

1. ACI Committee 318, Building Code Requirements for Structural Concrete (ACI 318-99) and Commentary (318R-99), American Concrete Institution, Farmington Hills, Michigan. (1999).


Application of Neural Networks for Modeling Shear Strength of Reinforced Concrete Beams


