Proceedings of the Institution of Civil Engineers
Structures & Buildings 157
August 2004 Issue SB4
Pages 279–292

Paper 13228
Received 09/01/2003
Accepted 10/10/2003

Keywords:
brickwork & masonry/design methods & aids/information technology

M. M. Reda Taha
Assistant Professor, Department of Civil Engineering, University of New Mexico, Albuquerque, NM, USA

A. Noureldin
Assistant Professor, Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, Ontario, Canada.

N. El-Sheimy
Associate Professor, Department of Geomatics Engineering, The University of Calgary, Calgary, AB, Canada

N. G. Shrive
Killam Memorial Research Chair, Department of Civil Engineering, The University of Calgary, Calgary, AB, Canada

Neural network modelling of creep in masonry

M. M. Reda Taha PhD, A. Noureldin PhD, N. El-Sheimy PhD and N. G. Shrive DPhil, FICE

Stresses and deformations in concrete and masonry structures can be significantly altered due to creep. However, accurate prediction of creep is difficult due to its dependency on a large number of parameters (e.g. section geometry, relative humidity, stress level, age of loading). This paper introduces a new method based on artificial intelligence to model creep of masonry. Feedforward artificial neural networks (ANN) are investigated as a modelling technique for predicting creep. Experimental data for creep of structural masonry are used to develop the networks. Changes in network architecture are examined to produce prediction models. Fifteen networks are developed and analysed statistically. Creep models with accuracy in the range ± 15% are attainable using ANN.

NOTATION

A	experimental constant for creep prediction models
B	experimental constant for creep prediction models
$\mathbf{b}$	bias vector
$\mathbf{c}_{\mathrm{sp}}$	specific creep strain
d_{j}	output desired value at layer j
E	mean square error
$E(t_0)$	modulus of elasticity of the materials at time of load application t_0
$F(t_0)$	experimental constant for creep prediction models
f	transfer function
$\mathbf{g}$	vector of gradient of the mean square error used in learning algorithms
I	transfer function net input
J	Jacobian matrix including the error gradients $(\partial E/\partial y)$
$J(t, t_0)$	creep compliance between time t_0 and t
m	number of samples in the testing set
N	number of hidden layers in the network
P	neuron input
PE	prediction error
R	number of neurons in a specific layer
t	time of creep prediction
t_0	age at time of load application
$\mathbf{W}$	weight matrix
x_{j}	input value at layer j
y_{i}	output activity value of previous layer i
y_{j}	output activity value at layer j
y_{pi}	predicted value
y_{ti}	measured value
$\boldsymbol{\varepsilon}(t_0)$	instantaneous strain at time t_0
$\boldsymbol{\varepsilon}_{\mathrm{cr}}(t, t_0)$	creep strain between time t_0 and t
$\boldsymbol{\varepsilon}_{\mathrm{sp}}$	specific elastic strain
μ_{k}	positive constants used to control the iterative changes during learning
ρ_{xy}	correlation coefficient between parameters x and y
$\boldsymbol{\sigma}$	sustained stress
ϕ_∞	experimental constant in creep prediction models
$\phi(t, t_0)$	creep coefficient between time t_0 and t
$\phi_{\mathrm{predictedANN}}\ (t, t_0)$	creep coefficient between time t_0 and t predicted using the ANN model
$\phi_{\mathrm{measured}}\ (t, t_0)$	creep coefficient between time t_0 and t measured experimentally

1. INTRODUCTION

Creep is the time-dependent, non-elastic strain that occurs in quasi-brittle materials (e.g. concrete and masonry) when load is sustained on a structure. It is well established that there is an inherent interplay between creep and shrinkage due to their interwoven mechanisms and their dependence on internal moisture movement.[1–2]

The consequence of creep depends on the level of restraint to the creep strain provided by the structure. If the deformation is not restrained, creep might result in a significant increase in deformation over time, possibly violating a serviceability limit state. When the creep strain is restrained, creep will induce new stresses in the structures which can alter the stress distribution and might result in local overstressing. Thus, the consequences of creep in structures have been the focus of several studies in the past few decades.[3–7]

Investigation of the collapse of the Civic Tower of Pavia in Italy in 1989 led to the conclusion that time-dependent deformation due to creep of the brickwork façades might have been a major reason for the collapse of this historical building.[8]

Dilger[6] and Mirmiran *et al.*[9] showed that restrained creep deformations of continuous concrete bridges can result in significant moment redistribution along the bridge elements.

Moreover, Reda Taha and Shrive[10] showed that when two materials (e.g. brickwork and grout) with different creep properties are used in a composite structural element, the effect of difference in creep of both materials can result in stress fluctuations which might lead to overstressing, cracking or failure of either material. On the other hand, creep can also have beneficial effects: for example, reducing the probability of cracking of high-performance concrete slabs by relieving the stresses developed due to restrained autogenous shrinkage.[11]

Therefore, in many instances accurate prediction of creep is necessary for efficient design, especially when materials with different creep behaviours are used in the structure (e.g. masonry, concrete, fibre reinforced polymers (FRPs) and grout). Numerous empirical models to predict creep of concrete and masonry using conventional functional mapping mathematics have been developed over the past 60 years.[12–15] The relative accuracy of these models has typically never been better than ± 15%.[15] This is because most of these models utilise curve-fitting techniques achieved by means of linear and non-linear regression analysis for specific sets of experimental data. The limited accuracy of conventional creep models is attributed to the dependency of creep on a large number of uncertain and interrelated parameters which makes curve-fitting techniques an oversimplified approach.

1.1. Mathematical representation of creep

Creep strain is usually related to its corresponding elastic strain. The specific instantaneous elastic strain, $\boldsymbol{\varepsilon}_{\text{sp}}$, that results from a given stress, $\boldsymbol{\sigma}$, is given in equation (1)

$$\boldsymbol{\varepsilon}_{\text{sp}} = \frac{\boldsymbol{\varepsilon}(t_0)}{\boldsymbol{\sigma}} \tag{1}$$

where $\boldsymbol{\varepsilon}(t_0)$ is the instantaneous elastic strain. Similarly the specific creep, c_{sp}, can be defined as in equation (2)

$$\boldsymbol{c}_{\text{sp}}(t,\ t_0) = \frac{\boldsymbol{\varepsilon}_{\text{cr}}(t,\ t_0)}{\boldsymbol{\sigma}} \tag{2}$$

where $\boldsymbol{\varepsilon}_{\text{cr}}(t, t_0)$ is the creep strain at time t due to application of the stress $\boldsymbol{\sigma}$ at time t_0. The ratio between the specific creep $\boldsymbol{c}_{\text{sp}}(t,t_0)$ and the specific elastic strain, $\boldsymbol{\varepsilon}_{\text{sp}}$, is equal to the ratio between the creep strain to the elastic strain. This ratio is known as the creep coefficient $\phi(t, t_0)$.

$$\phi(t,\ t_0) = \frac{\boldsymbol{c}_{\text{sp}}(t,\ t_0)}{\boldsymbol{\varepsilon}_{\text{sp}}} \tag{3}$$

The creep coefficient ϕ (t, t_0) is used by many design codes[16, 17] to estimate long-term creep stresses and deformation. However, the creep compliance $J(t, t_0)$ has been recommended by RILEM Technical Committee TC 107 for comparing the accuracy of different creep models.[18] The $J(t, t_0)$ between the time of loading t_0 and any time t is

$$J(t,\ t_0) = \frac{1}{E(t_0)} + \boldsymbol{c}_{\text{sp}}(t,\ t_0) \tag{4}$$

where $E(t_0)$ is the material modulus of elasticity at time of load application t_0.

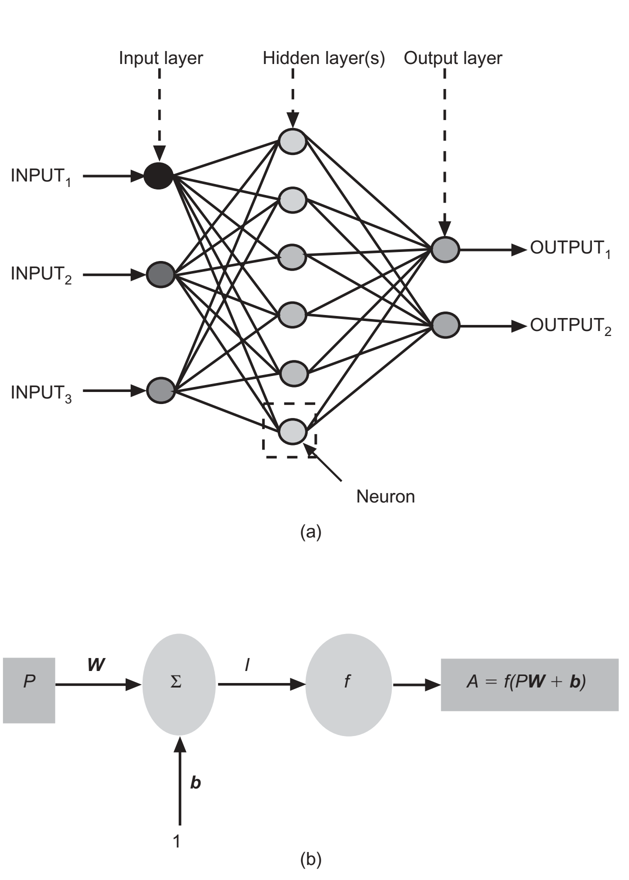


Fig. 1. Artificial neural networks: (a) sample network structure; (b) operation at the neuron level

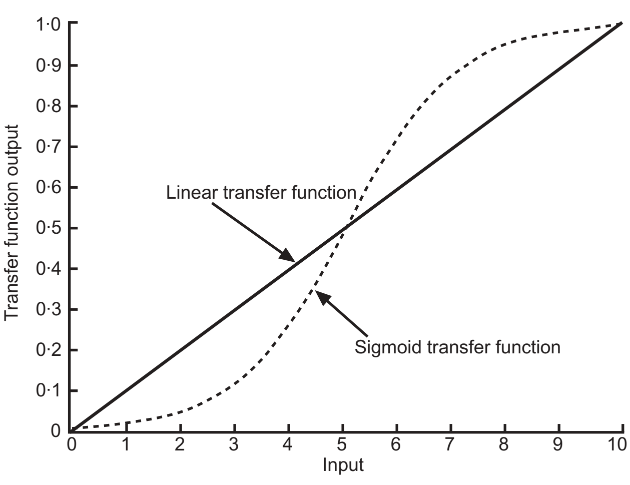


Fig. 2. Sigmoid and linear transfer functions

Group	Age of loading: days	Stress level: MPa	RH: %
Training 1	7	2·4	40
Training 2	7	4·8	100
Training 3	14	4·8	40
Training 4	14	4·8	100
Training 5	28	1·2	40
Training 6	28	2·4	40
Training 7	28	4·8	40
Training 8	28	2·4	100
Training 9	28	3·6	100
Training 10	28	4·8	100

Table 1. Experimentally measured creep groups used in training ANN models

Neville *et al.*[19] grouped creep prediction models into two categories: the first category includes those models with a maximum threshold using exponential and hyperbolic expressions as in equations (5) and (6)

Group	Age of loading: days	Stress level: MPa	RH: %
Testing 1	7	4·8	100
Testing 2	14	2·4	40
Testing 3	28	1·2	100
Testing 4	28	3·6	40

Table 2. Experimentally measured creep groups used in testing ANN models

$$\phi(t,\ t_0) = E(t_0)\left(\frac{t - t_0}{A + B(t - t_0)}\right) \text{ exponential expressions} \quad (5)$$

$$\phi(t,\ t_0) = \phi_\infty(1 - \mathrm{e}^{-A(t-t_0)}) \text{ hyperbolic expressions} \quad (6)$$

The second category includes those models that increase indefinitely using logarithmic and power expressions as presented in equations (7) and (8)

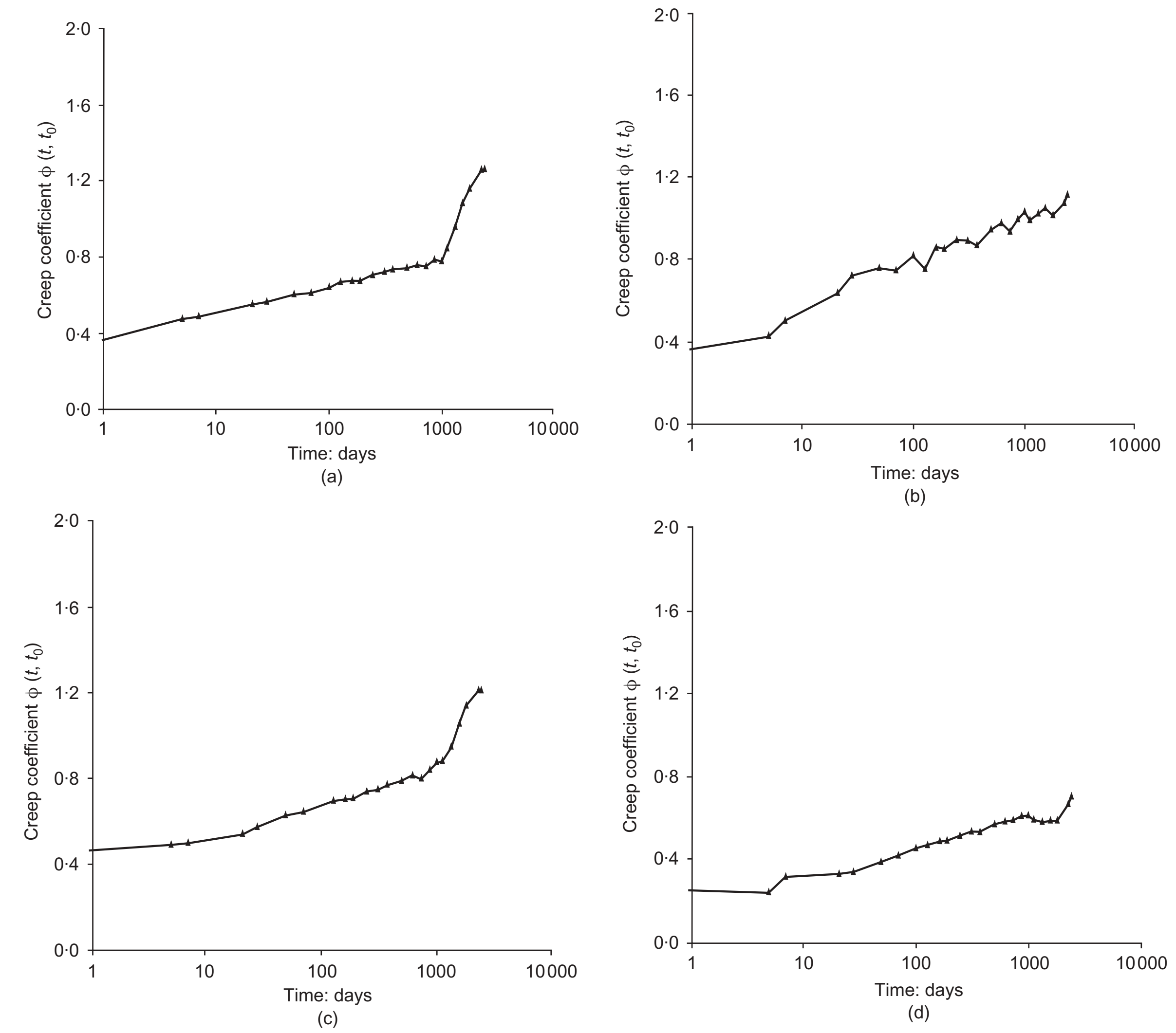


Fig. 3. Sample of experimental creep data used in training the ANN models, training groups: (a) 2; (b) 4; (c) 7; (d) 10

No.	Network denotation	N	R_1	R_2	Transfer function (1)	Transfer function (2)
1	RTN-1G	1	1	—	log-sigmoid	—
2	RTN-2G	1	2	—	log-sigmoid	—
3	RTN-3G	1	3	—	log-sigmoid	—
4	RTN-4G	1	4	—	log-sigmoid	—
5	RTN-6G	1	6	—	log-sigmoid	—
6	RTN-8G	1	8	—	log-sigmoid	—
7	RTN-10G	1	10	—	log-sigmoid	—
8	RTN-64GG	2	6	4	log-sigmoid	log-sigmoid
9	RTN-64GL	2	6	4	log-sigmoid	pure-linear
10	RTN-33GG	2	3	3	log-sigmoid	log-sigmoid
11	RTN-24GL	2	2	4	log-sigmoid	pure-linear
12	RTN-24GG	2	2	4	log-sigmoid	log-sigmoid
13	RTN-24LL	2	2	4	pure-linear	pure-linear
14	RTN-46GG	2	4	6	log-sigmoid	log-sigmoid
15	RTN-46GL	2	4	6	log-sigmoid	pure-linear
16	Modified Burgers	—	—	—	—	—

N: number of layers
R_1: number of neurons in layer 1
R_2: number of neurons in layer 2

Table 3. ANN investigation parameters, including the number of hidden layers, N, the number of neurons of each layer, R_1 and R_2, and the transfer function in each layer

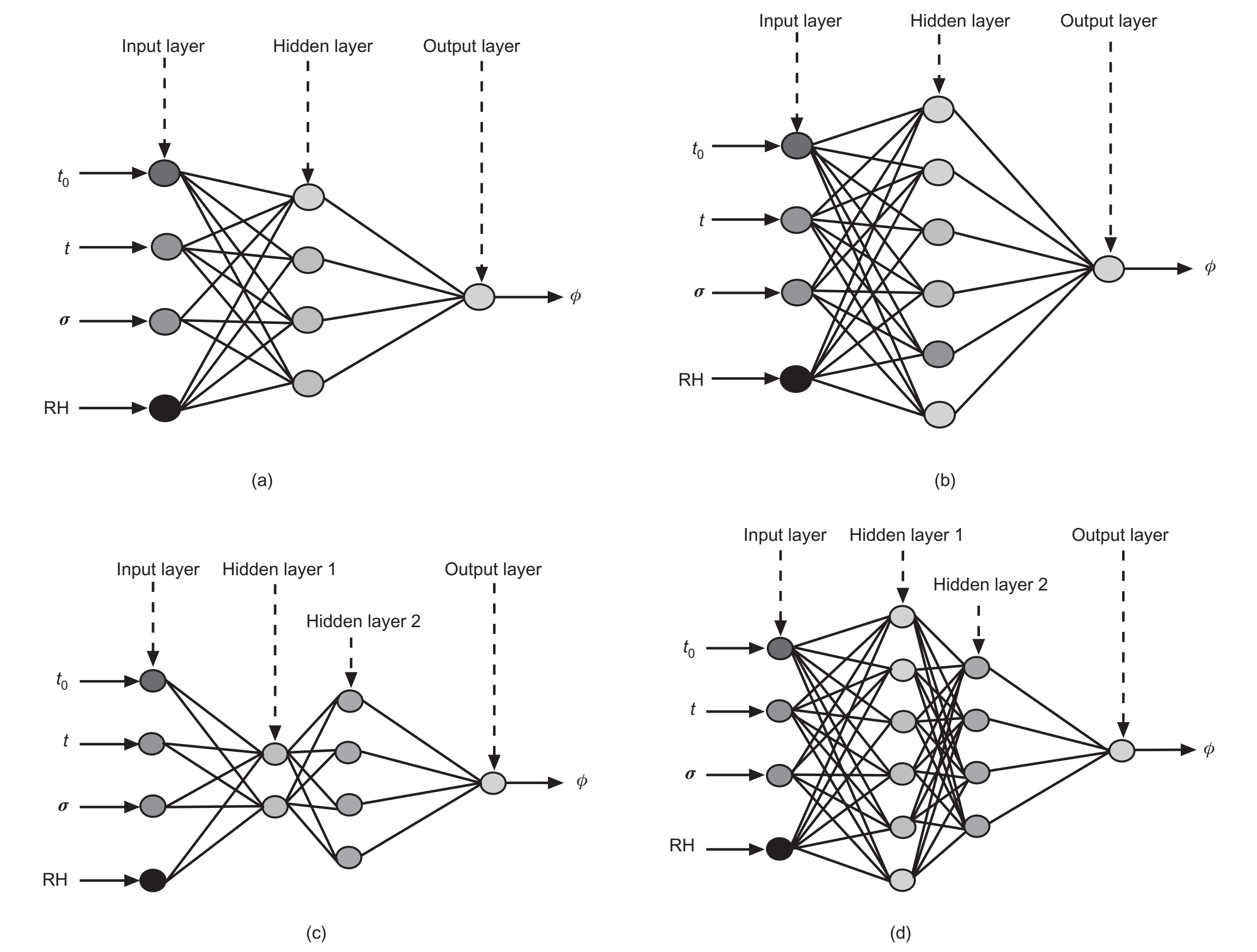


Fig. 4. Architecture of sample of the artificial neural networks (ANN) developed for predicting creep: (a) RTN-4G; (b) RTN-6G; (c) RTN-24GL; (d) RTN-64GL

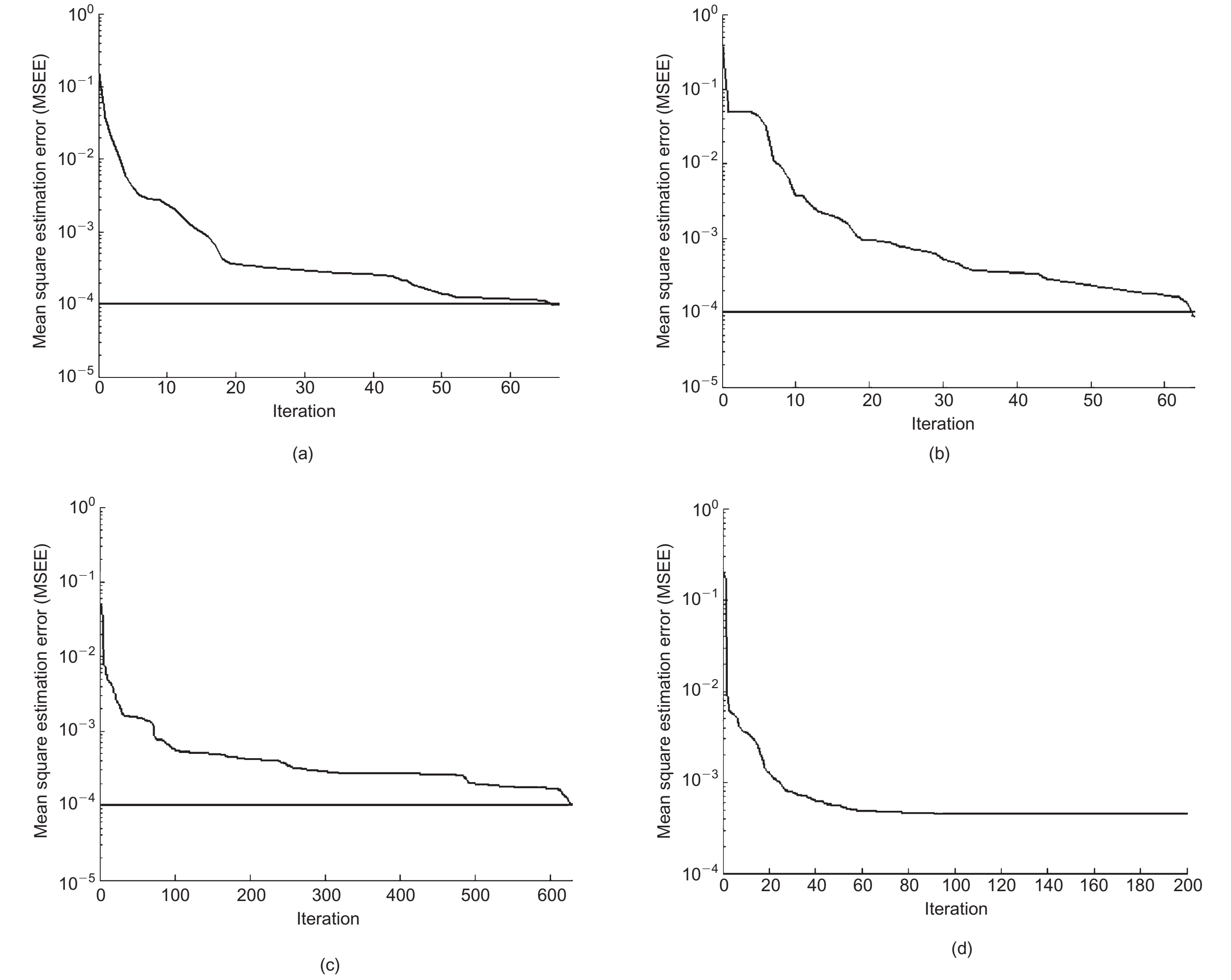


Fig. 5. Convergence of sample networks during training (target accuracy 0·0001): (a) RTN-6G; (b) RTN-8G; (c) RTN-64GG; (d) RTN-33GG

7	$\phi(t,\ t_0) = E(t_0)F(t_0)\log(t - t_0 + 1)$ logarithmic expressions

8	$\phi(t,\ t_0) = E(t_0)A(t - t_0)^B$ power expressions

The constants (A, B, $F(t_0)$, ϕ_∞, etc.) in equations (5)–(8) need to be determined experimentally using regression analysis. Although most of these previous models can predict creep for a few sets of experimental data, they have not proven accurate as general models.

The inability of these models to simulate creep behaviour under gradually increasing or decreasing loads or to predict accurately the effect of partial unloading on the creep strains has directed researchers to develop models that utilise rheological analogy. Rheological models are capable of providing comprehensive modelling of creep[20] but have rarely been used in design.

2. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANN) are an artificial intelligence tool that were first proposed more than two decades ago for modelling systems that have complex non-linear input–output relationships. Neuron computing, a technology of ANN, is a powerful tool for solving such non-linear problems that involve mapping input data to output data without having any prior knowledge about the mathematical process involved. ANN consist of densely interconnected processing units that utilise parallel computation algorithms. The basic advantage of ANN is that they can learn from representative examples from the data set.[21,22] While ANN do not provide a closed form mathematical model for the problem, they do offer accurate models based on the learning procedure.

2.1. ANN architecture

ANN are networks of many simple processors (neurons) operating in parallel, each possibly having a small amount of local memory. A neural network consists of an input layer, an output layer and number of layers between the input and the output layers known as hidden layers. A representative sample of ANN architecture is shown in Fig. 1(a), consisting of an input layer with three input parameters, an output layer with

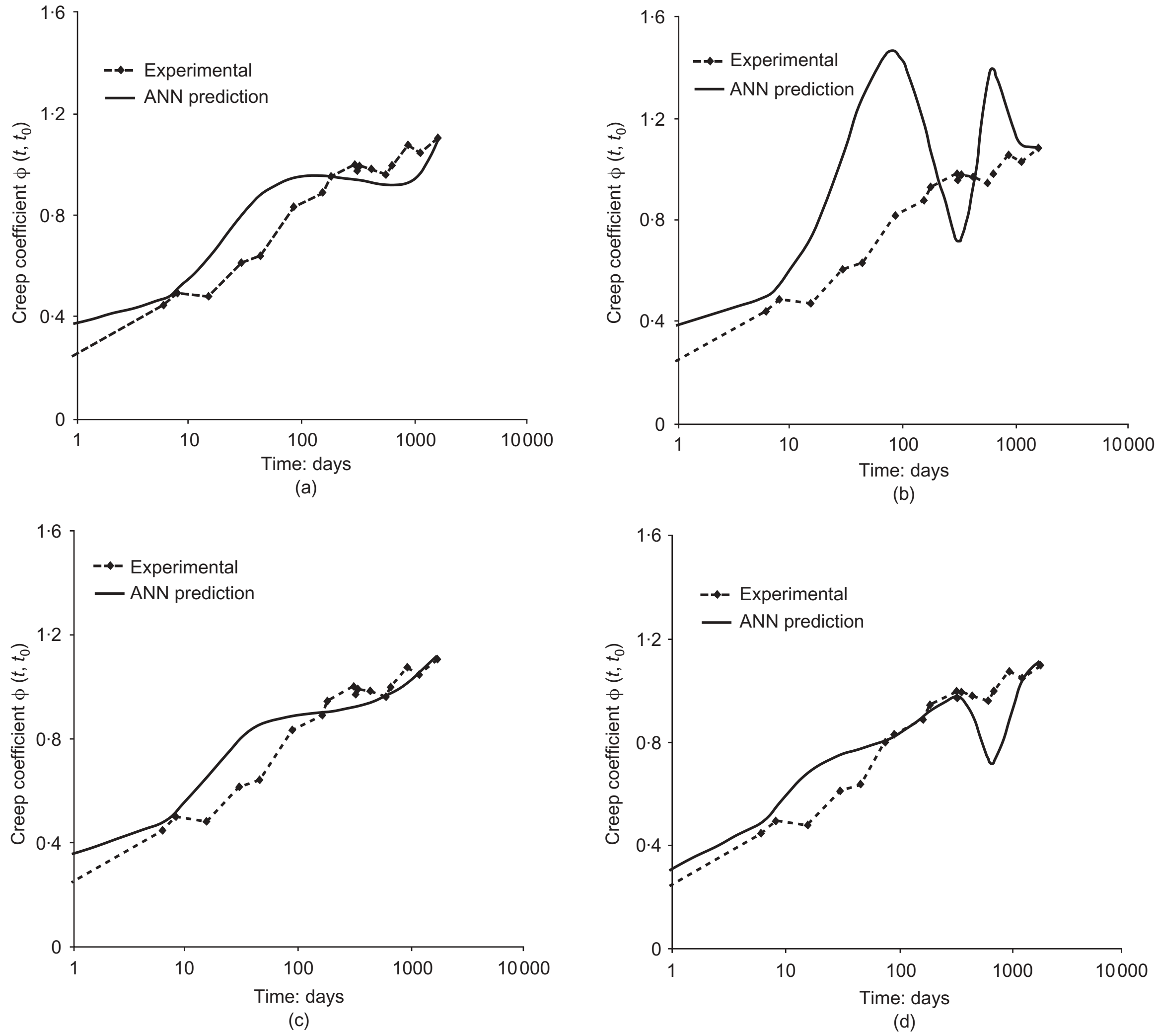


Fig. 6: Samples of creep prediction using ANN versus measured creep coefficient $\phi(t, t_0)$ (measured creep for testing group 4): (a) RTN-6G; (b) RTN-8G; (c) RTN-64GG; (d) RTN-33GG

two output parameters and a single hidden layer with six neurons. The number of neurons in the input and the output layers of any network has to be equal to the number of the inputs and outputs of the system respectively. The number of neurons in the hidden layers and the number of hidden layers can be arbitrarily chosen and adjusted until the network can map the desired output.[22, 23] However, it has been reported that one or two hidden layers with an arbitrary number of neurons in each layer can model virtually any non-linear input–output relationship.[24]

The smallest network unit (the neuron) receives its input through a connection that multiplies the value of the input by a scalar weight, *W*, and adds a bias, ***b***. The sum of the weighted inputs and their weights and biases is the argument for a transfer function, *f*, that produces the neuron output. The operation at the neuron level is shown in Fig. 1(b). The pattern of connectivity in the network is represented by a weight vector, ***W***. The initial values for the weights and biases of the network can be arbitrarily chosen. By adjusting the ***W*** and the ***b*** the network can exhibit any desired output. The process of adjusting the weights and the biases of the network is known as training. In other words, an ANN learns from examples (of known input–output sequences) and exhibits some capability for generalisation beyond the training data.[24]

2.2. Transfer functions

Transfer functions for the neurons are needed to introduce non-linearity into the network. Bounded transfer functions such as logistic functions are particularly useful when the target values have a bounded range. However, if the target values have non-bounded ranges, it is preferable to use an unbounded transfer function. Transfer functions commonly used in feedforward neural networks include linear, sigmoid and log-sigmoid transfer functions.[22–24] These transfer functions have outputs ranging between 0 and 1 and are suitable for backpropagation networks (explained below) because they are differentiable.[22]

A comparison between the input–output relationships of both sigmoid and linear transfer functions which are used in the networks described below are shown in Fig. 2. Networks of two layers that utilise sigmoid and linear transfer functions in the

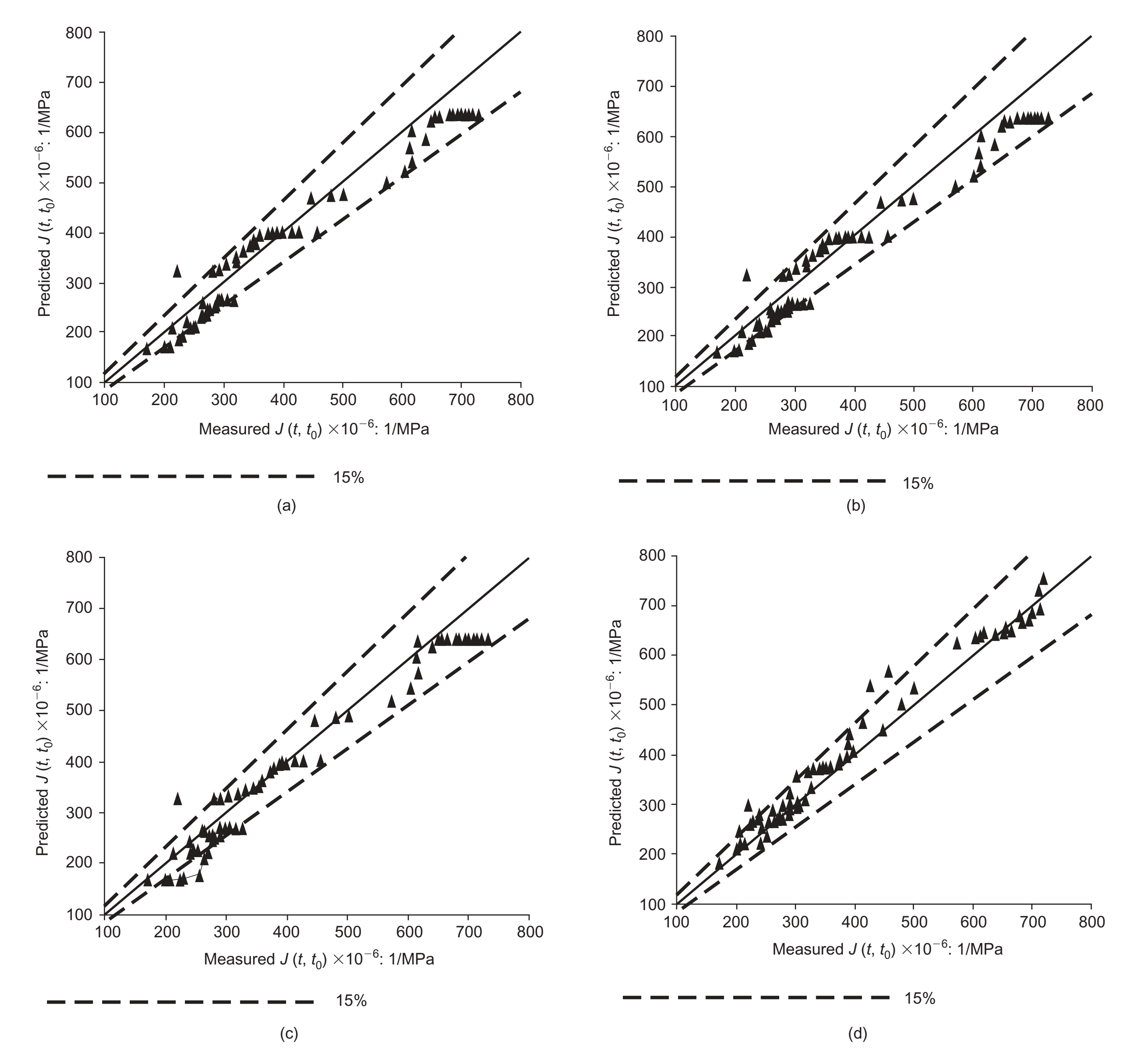


Fig. 7: ANN models for predicting creep compliance: ANN model 1 to 4: (a) RTN-1G; (b) RTN-2G; (c) RTN-3G; (d) RTN-4G

first and second layer respectively can be trained to model any non-linear relation.[22, 23]

2.3. Learning rules and backpropagation algorithm

The learning rule (or training algorithm) is a procedure for modifying the weights and biases of the network. Learning rules fall into two broad categories: supervised learning and unsupervised learning. In supervised learning, the learning rule is provided with a known input/output set of data and an algorithm is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets. Therefore, modelling capabilities of networks trained using supervised learning algorithms are limited to the range of input data used in training the network. In unsupervised learning the weights and biases of the network are modified according to the inputs only. Unsupervised learning is usually used in data partitioning.[23–25]

The basic learning rule in feedforward networks is the gradient descent method which is a classic technique for minimising a given function defined on a multidimensional input space. The gradient descent method requires finding a gradient vector, $\boldsymbol{g}$, in which each element is defined as the derivative of an error measure with respect to a network parameter. The procedure for finding this $\boldsymbol{g}$ is known as backpropagation because the gradient vector is calculated in a direction opposite to the flow of data in the network.[25, 26]

Therefore, the task of the backpropagation algorithm is to minimise the overall error measure of the network so that the network prediction matches the desired output. There is no single universal error measure that is suitable for use in all networks. However, the sum of the squared errors, E, presented in equation (9) is the most commonly used error measure

$$E = \sum_{i=1}^{N} (d_i - y_i)^2 \tag{9}$$

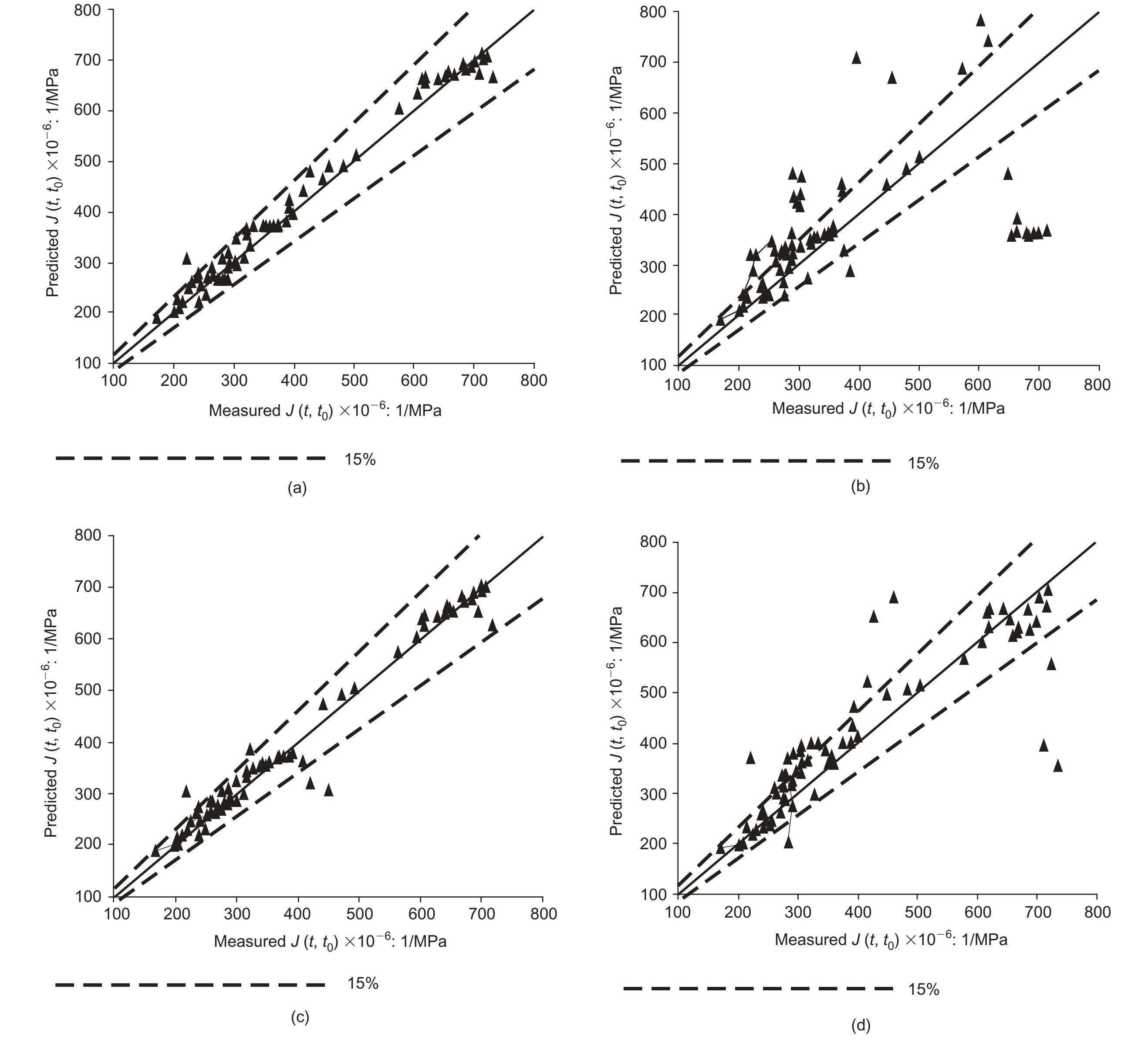


Fig. 8: ANN models for predicting creep compliance: ANN model 5 to 8: (a) RTN-6G; (b) RTN-8G; (c) RTN-10G; (d) RTN-64G

where y_i is the ith element in the network output vector and d_i is the ith element in the desired output vector and N is the total number of outputs predicted by the network. The error measure E is minimised by altering the $\boldsymbol{W}$ and $\boldsymbol{b}$ so that the desired output is achieved by the network. In-depth discussion about the mathematical bases of the backpropagation algorithm is beyond the scope of this work but can be found in almost all neural network textbooks.[22–27]

3. DEVELOPMENT OF ANN FOR PREDICTING CREEP

The aim of this paper was to investigate the potential of using feedforward neural networks in predicting creep of quasi-brittle materials and to examine the effect of changing the network architectural parameters on the network performance. Creep data extracted from experiments on structural masonry prisms collected continuously over the past 15 years by Shrive and Tilleman.[28] have been used for providing the training and testing data sets needed for the development of the ANN for predicting creep.

3.1. Experimental data

All the masonry prisms were made of 90 × 190 × 57 mm standard bricks and standard type N mortar (1 Portland cement : 1 lime : 6 sand). The specimens were kept under two environmental conditions: the first group was sealed and kept continuously wet by providing an outer source of water to the specimen (RH = 100%), while the second group was kept in the room humidity (RH = 40%). All specimens were kept in a laboratory at a temperature of 20 ± 2°C.

The creep deformations of the masonry prisms subjected to different stress levels (1·21, 2·43, 3·61 and 4·86 MPa) representing 12, 24, 36 and 48% of the prisms' compressive strength respectively and exposed to different environmental conditions were used to develop and assess the model.[7,28] To

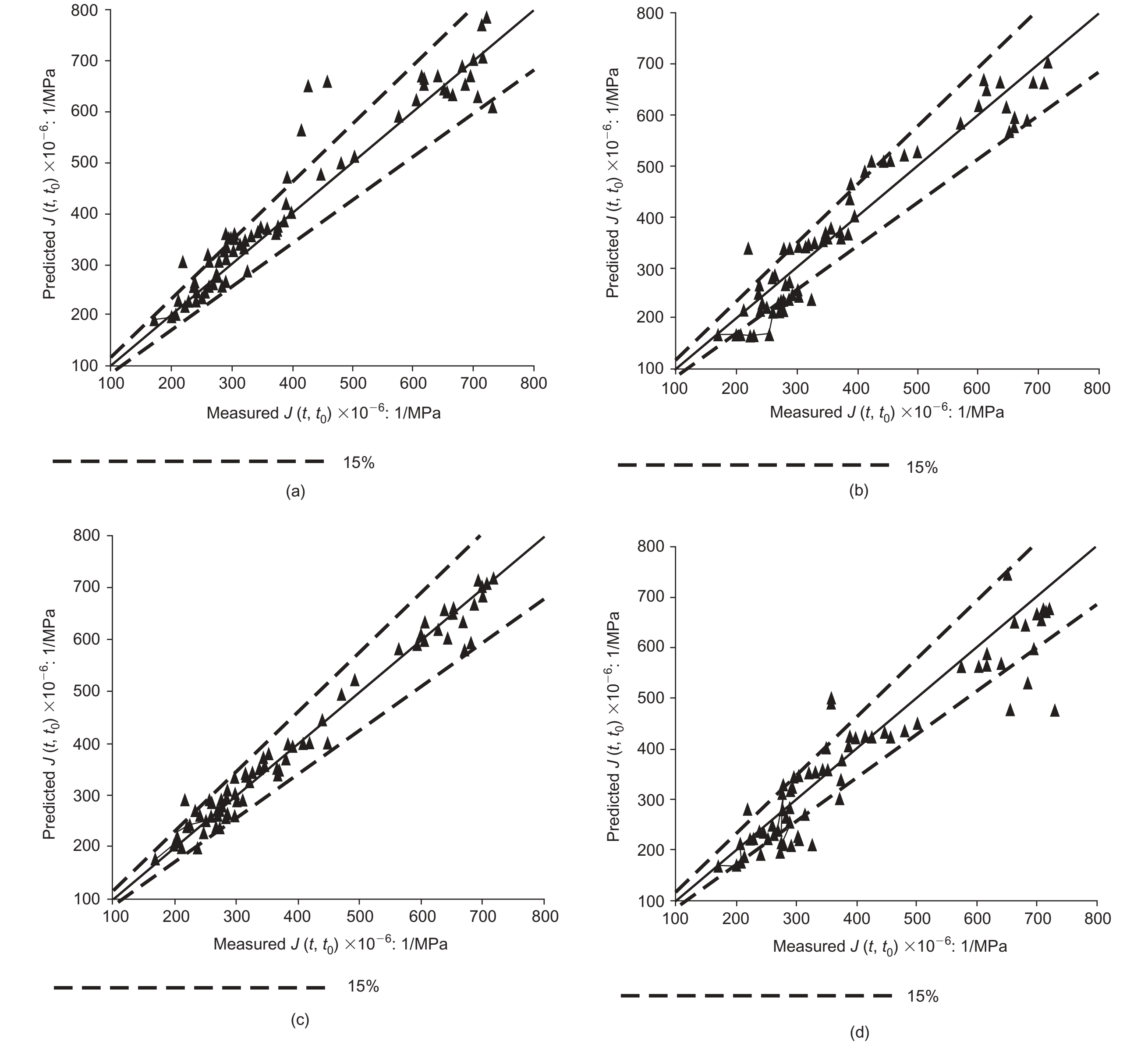


Fig. 9: ANN models for predicting creep compliance: ANN model 6 to 9: (a) RTN-64GL; (b) RTN-33GG; (c) RTN-24GL; (d) RTN-24GG

compensate for the effects of shrinkage and temperature, strains of unloaded prisms subjected to environmental conditions similar to their counterpart-loaded prisms were also recorded. Fourteen experimental testing groups were used for training and testing the networks as presented in Table 1 (ten training groups) and Table 2 (four testing groups).

The creep coefficient ϕ (t, t_0) and the creep compliance $J(t, t_0)$ were determined from the data using equations (1)–(4). Samples representing the change of the creep coefficient with time for four training groups are shown in Fig. 3.

As all the experiments were performed on specimens of the same size, four parameters only were considered for modelling creep: the applied stress level, $\boldsymbol{\sigma}$, the relative humidity (RH), the age of loading, t_0, and the time at which creep is measured, t. The effect of temperature on creep was not examined here. Also, the surface area to volume ratio was not considered as a changing parameter, being constant for all tests. Preliminary investigations showed that the inclusion of constant values representing the surface area and the temperature would not have any effect on the performance of the ANN.

3.2. Training algorithm

Fifteen feedforward ANN for modelling creep deformations of structural masonry were developed. The 15 networks consist of an input layer with four neurons, N hidden layers of R neurons and an output layer with one neuron. The number of hidden layers, N, and neurons, R, for each network are given in Table 3. The efficiencies of both the pure-linear transfer function and the log-sigmoid transfer function were also examined. The transfer functions used in each layer of the networks are listed in Table 3. A schematic of sample architecture of four of the ANN developed for modelling creep is presented in Fig. 4.

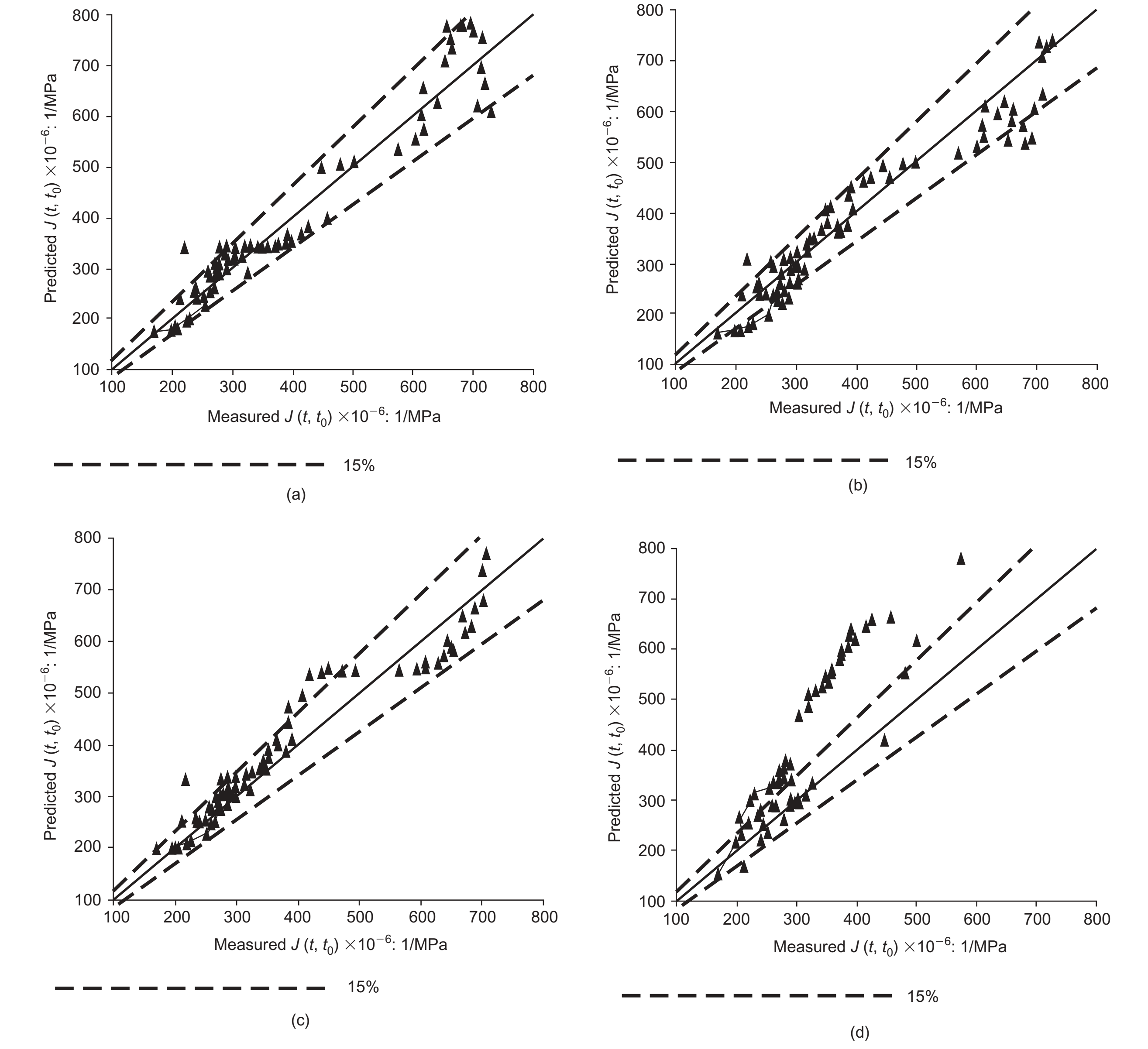


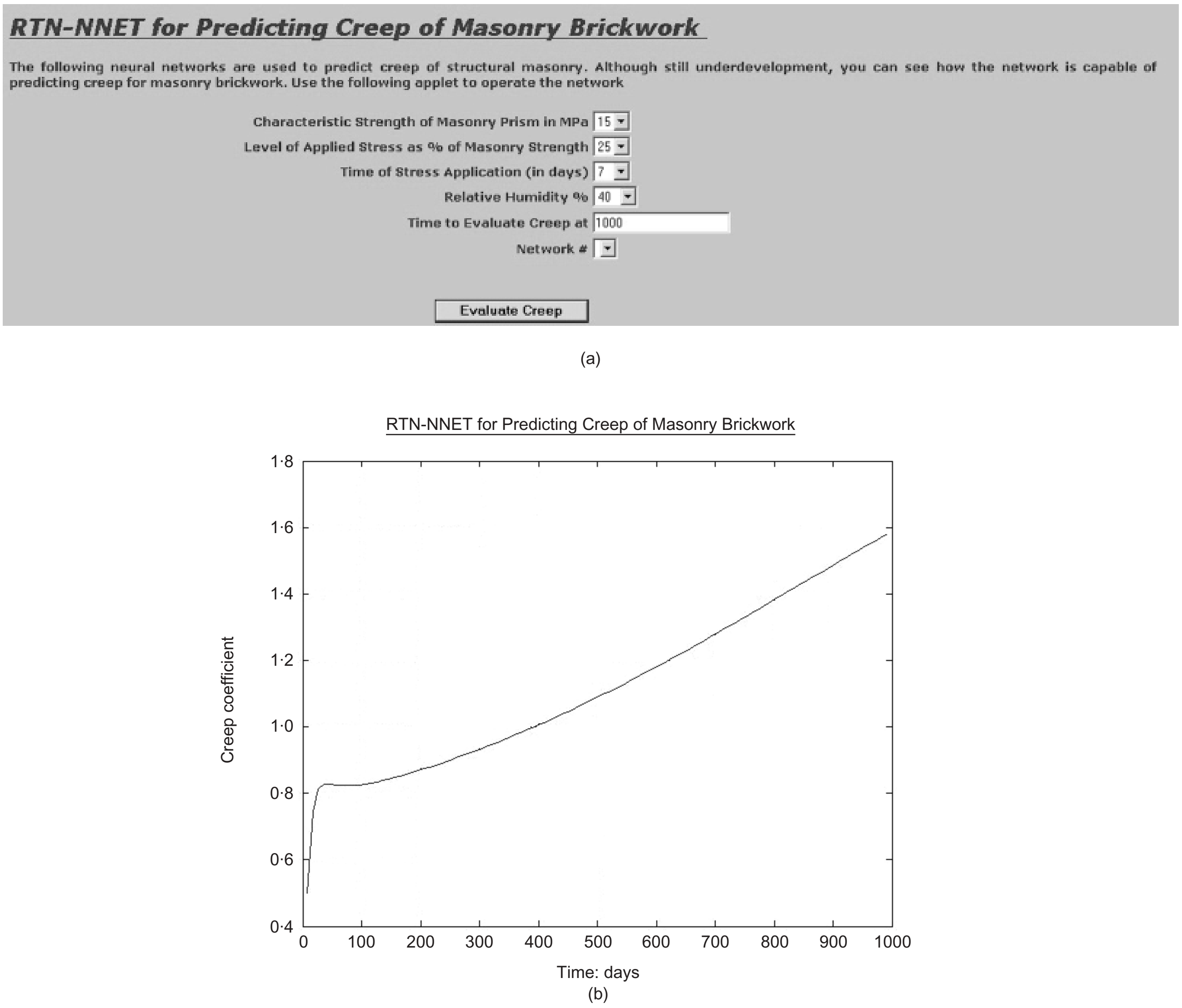
Fig. 10: ANN models for predicting creep compliance: ANN model 10 to 15 and modified Burgers model: (a) RTN-2455; (b) RTN-4499; (c) RTN-469L; (d) modified Burgers

All 15 networks utilise the backpropagation training algorithm as the learning rule for the network with the Levenberg-Marquardt weights update criterion.[26] This criterion is based on the gradient descent method described above but with a small modification that speeds up the training procedure and provides minimal mean square estimation error (MSEE).[26, 27] This training algorithm was implemented using the Neural Network Toolbox of MATLAB® computer-aided design software.[29] A learning matrix including 47 training samples drawn from the ten training groups was used in training each network. In order to achieve fast convergence to the target MSEE of (1×10^{-4}), the input and output data were normalised with respect to the corresponding maximum values in the input vectors using linear normalisation functions. All 15 networks successfully achieved the target mean square estimation error (MSEE) except network No. 4 10 (RTN-33GG) which achieved an MSEE of ($4{\cdot}5 \times 10^{-4}$) as shown in Fig. 5(d). Convergence of the MSEE of four sample networks during training is presented in Fig. 5. The structure of the network and its transfer functions clearly affect the number of iterations needed during the training procedure of the network to achieve the target MSEE. While RTN-6G (Fig. 5(a)) was able to achieve the target MSEE after 70 iterations, RTN-64GG (Fig. 5(c)) was able to achieve this target MSEE after 650 iterations and RTN-33GG (Fig. 5(d)) was not able to achieve the specified MSEE as mentioned above. The number of iterations also represents the time needed for network training.

4. TESTING THE ANN MODELS FOR PREDICTING CREEP

4.1. Comparison with experimentally measured creep

Each of the 15 networks was tested using a matrix of 80 samples drawn from the four testing groups presented in Table 2. These groups were not used in training the networks. The ANN models were used to predict the creep coefficient, $\phi(t, t_0)$. A sample creep prediction for four ANN versus measured creep coefficient $\phi(t, t_0)$ is presented in Fig. 6.



RTN-NNET for Predicting Creep of Masonry Brickwork

The following neural networks are used to predict creep of structural masonry. Although still underdevelopment, you can see how the network is capable of predicting creep for masonry brickwork. Use the following applet to operate the network

Characteristic Strength of Masonry Prism in MPa: 15

Level of Applied Stress as % of Masonry Strength: 25

Time of Stress Application (in days): 7

Relative Humidity %: 40

Time to Evaluate Creep at: 1000

Network #:

Fig. 11. Predicting creep on the worldwide web (http://cfei.geomatics.ucalgary.ca/matlab/ann.html): (a) RTN_NNET interface to predict creep; (b) creep prediction using the RTN web services

To examine the difference between the ANN prediction and creep prediction models developed using curve-fitting techniques, the experimentally measured creep was also compared to the modified Burgers model[7] as presented in equations (10) and (11)

$$\phi(t,\ t_0) = Rt^{0{\cdot}3} + (1 - \mathrm{e}^{-Rt}) \tag{10}$$

$$R = 0{\cdot}112 - 3{\cdot}35 \times 10^{-6} E(t_0) \tag{11}$$

Comparison of the creep compliance as predicted by the ANN models against the creep compliance determined experimentally is presented in Figs 7 – 10. To represent the accuracy of prediction of the networks visually, two dashed lines representing the ± 15% deviation from the 45° line of the predicted against measured creep compliance graph have been drawn.

Eight of the 15 networks were able to predict creep compliance within the ± 15% (RTN-1G, RTN-4G, RTN-6G, RTN-10G, RTN-33GG, RTN-24GL, RTN-24LL and RTN-46GG). The other seven networks and the modified Burgers model were not able to achieve this accuracy. Relative accuracies ranging from ± 30 to ± 50% are attainable using those networks and the classical regression analysis.

4.2. Statistical analysis

The large number of parameters examined in this study and the relatively large uncertainty in creep parameters make comparisons between individual ANN predicted creep values and experimentally measured creep inaccurate. Cross-validation techniques based on statistical values rather than absolute values are thus needed for evaluating ANN prediction.

A common cross-validation technique is to determine the correlation coefficient between measured and predicted parameters. The correlation coefficient between two parameters x and y, ρ_{xy} represents how closely the two parameters x and y are related. The correlation coefficient can describe the relationship between the trends of the prediction model and the experimental data rather than providing a solid statistical inference of how each model fits the experimental data.[30]

Therefore, the correlation coefficient was not used here to measure the efficiency of the prediction models. Rather, statistical comparisons between predicted and measured creep were performed by estimating the prediction error (PE) which measures the average squared error between the predicted creep obtained from the model and the measured creep. The PE is described in equation (12)

$$PE = \frac{1}{m}\sum_{i=1}^{m}(y_{\text{ti}} - y_{\text{pi}})^2 \tag{12}$$

where y_{pi} is the predicted value and y_{ti} is the experimentally measured value and m represents the number of samples in each testing group.[31] Prediction errors for the 15 networks and the modified Burgers model are presented in Table 4. It is obvious from Table 4 that creep prediction models using ANN have a smaller prediction error and consequently higher accuracy (e.g. RTN-1G, RTN-6G and RTN-33GG) than classical creep prediction models using conventional regression analysis (e.g. modified Burgers model).

An interesting observation was that accuracy of the ANN model does not necessarily increase with complexity of the network. On the contrary, simple networks with one hidden layer including one or six neurons (RTN-1G and RTN-6G) showed a relatively smaller prediction error ($PE < 0{\cdot}12$) than complex networks with two hidden layers including a large number of neurons (e.g. RTN-64GG, $PE > 0{\cdot}22$).

Another interesting observation is that the effect of the transfer function is as important as the number of layers and neurons in each layer. This can be observed when comparing the performance of two networks with similar numbers of hidden layers and neurons but with different transfer functions (e.g. RTN-46GG v. RTN-46GL and RTN-24GG v. RTN-24 GL). Networks including non-linear transfer functions do not necessarily yield accurate prediction and required longer training time than networks including linear transfer functions. The choice of the most suitable combination of transfer functions for a specific architecture of the network is therefore necessary for accurate modelling.

5. THE USE OF ANN CREEP PREDICTION MODEL IN STRUCTURAL DESIGN

The ANN creep prediction models presented here were developed for use in structural design via the world wide web. The ANN creep models are integrated using HTML computer language and invoked over the web for structural engineers worldwide. The networks can be accessed via the world wide web, for example currently at (http://cfei.geomatics.ucalgary.ca/matlab/ann.html). Fig. 11(a) shows the user interface of the RTN_NNET web program for predicting creep of structural masonry.

A structural engineer using the RTN_NNET on the web does not need to have prior knowledge of neural networks and their development but will need to know the basic design parameters that affect creep including the characteristic strength of the masonry, the stress level applied to the brickwork structure, the time of loading, the average relative humidity and the time for evaluating creep. The software output is the $\phi(t, t_0)$ evaluated for the whole time interval (t–t_0) as shown in Fig. 11(b). The advantage of using the creep prediction models with neural networks over current computer-based models is the ability to enhance the model with time as new test data become available.

6. CONCLUSION

The conclusions of this study can be summarised as follows.

(*a*) While accurate prediction of creep deformation is desired to increase the level of confidence in serviceability analysis, it is evident that a high level of accuracy cannot be accomplished with classical curve-fitting techniques due to the large number of parameters needed in the model.

No.	Model	Prediction error (PE)*			
		Group 1	Group 2	Group 3	Group 4
1	RTN-1G	0·22	0·19	0·23	0·18
2	RTN-2G	0·24	0·84	0·25	0·54
3	RTN-3G	0·19	0·15	0·28	0·17
4	RTN-4G	0·11	0·10	0·12	0·09
5	RTN-6G	0·11	0·07	0·10	0·07
6	RTN-8G	0·62	1·04	0·32	0·87
7	RTN-10G	0·15	0·08	0·06	0·28
8	RTN-64GG	0·29	0·34	0·22	0·28
9	RTN-64GL	0·27	0·14	0·80	0·23
10	RTN-33GG	0·26	0·29	0·34	0·08
11	RTN-24GL	0·16	0·10	0·12	0·17
12	RTN-24GG	0·34	0·29	0·29	0·23
13	RTN-24LL	0·15	0·19	0·15	0·25
14	RTN-46GG	0·16	0·22	0·27	0·09
15	RTN-46GL	0·12	0·20	0·13	0·11
16	Modified Burgers	0·13	0·70	0·51	0·75

* PE: is dimensionless as it discribes the prediction error of the creep coefficient $\phi(t,t_0)$ which is dimensionless

Table 4. Prediction error (PE) between predicted and experimentally measured creep

(*b*) Fifteen ANN models were developed to investigate the potential use of feedforward neural networks in predicting time-dependent creep deformations in masonry structures. Experimental data on creep of structural masonry were used to develop the networks.

(*c*) Eight networks were able to predict creep with high levels of relative accuracy in the range ± 15%. Statistical analysis of these networks showed a relatively small prediction error compared to those associated with the other networks as well as with the classical model based on regression analysis.

(*d*) It can be observed that the change of the transfer function in ANN is as significant as the change of the network architecture including the number of layers and number of neurons.

(*e*) The neural networks introduced here can be used for predicting creep during structural design of a masonry structures via the world wide web. The ANN creep prediction model is currently available at (http://cfei.geomatics.ucalgary.ca/matlab/ann.html). The use of the ANN models as web application has the privilege of allowing dynamic development and enhancement of the model as further test data become available.

7. ACKNOWLEDGEMENTS

The financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged. The authors would like to extend their thanks to the technical staff at the University of Calgary for obtaining the creep data. Thanks also to Fabiola Narvaez for her useful comments on the first draft of this paper.

REFERENCES

1. Kovler K. Drying creep of stress-induced shrinkage? *Proceedings of Creep, Shrinkage and Durability Mechanics of Concrete and Other Quasi-brittle Materials* (Ulm F.-J., Bazant Z. P. and Wittmann F. H. (eds)). Elsevier, Oxford, 2001, pp. 67–72.
2. Brooks J. J. A theory of drying creep of concrete. *Magazine of Concrete Research*, 2001, **53**, No. 1, 51–61.
3. Lencczner D. Creep in model brickwork. *Proceedings of Designing Engineering and Construction with Masonry Products* (Johnston F. B. (ed.)), Houston, Texas, 1969, pp. 1958–1969.
4. Brooks J. J. and Abdullah C. S. Composite modelling of the geometry influence on creep and shrinkage of calcium silicate brickwork. *Proceedings of the British Masonry Society*, 1969, **4**, 36–38.
5. Ameny P., Jessop E. L. and Loov R. E. Strength, elastic and creep properties of concrete masonry. *International Journal of Masonry Construction*, 1980, **1**, No. 1, 33–39.
6. Dilger W. H. Creep analysis of reinforced concrete structures. *American Concrete Institute* (special publication), 1982, **SP-76**, 325–339.
7. Shrive N. G, Sayed-Ahmed E. Y. and Tilleman D. Creep analysis of clay masonry assemblages. *Canadian Journal of Civil Engineering*, 1997, **24**, No. 3, 367–379.
8. Binda L., Gatti G., Mangano G., Poggi C. and Sacchi Landriani G. The collapse of the civic tower of Pavia: a survey of the materials and structure. *Masonry International*, 1992, **6**, No. 1, 11–20.
9. Mirmiran A., Kulkarni S., Miller R. and Hastak M. Nonlinear continuity analysis of precast, prestressed concrete girders with cast-in-place decks and diaphragms. *PCI Journal*, 2001, **46**, No. 5, 61–77.
10. Reda Taha M. M. and Shrive N. G. Numerical modelling of the non-linear creep effects on the stress distribution in composite structures. *Proceedings of the 9th Arab Structural Conference*, UAE, 2003, 47–53.
11. Kanstad T., Bjøntegaard Ø, Sellevold E. J., Hammer T. A. and Fidjestøl P. Effects of silica fume on crack sensitivity. *Concrete International*, 2001, **23**, No. 12, 53–59.
12. Troxell G. E., Raphael J. M. and Davis R. E. Long-time creep and shrinkage tests of plain and reinforced concrete. *ASTM Proceedings*, 1958, **58**, 1101–1120.
13. Jordaan I. J., England G. L. and Khalifa M. A. Creep of concrete, a consistent engineering approach. *ASCE Journal of the Structural Division*, 1977, **103**, ST No. 3, 475–491.
14. Van Zijl G. P. A. G. A numerical formulation for masonry creep, shrinkage and cracking. In *Series 11—Engineering Mechanisms 01*. Delft University Press, Delft, Netherlands, 1999.
15. Gardner J. and Lockman M. J. Design provisions for drying shrinkage and creep of normal strength concrete. *ACI Materials Journal*, 2001, **98**, No. 2, 159–167.
16. CSA-1994 Masonry Code. *Masonry Design for Buildings (Limit States Design)—Structures (Design)*. CSA, Ontario, 1994, CSA-S304·1-94.
17. CEP-FIP Model Code 90. *Model Code for Concrete Structures*. Comité Euro-International du Beton (CEB)–Fédération Internationale de la Précontrainte (FIP), Thomas Telford, London, 1993.
18. RILEM Technical Committee. Guidelines for characterizing concrete creep and shrinkage in structural design codes or recommendations, TC 107: creep and shrinkage prediction models: principles of their formulation. *Materials and Structures*, 1995, **28**, 52–55.
19. Neville A. M., Dilger W. H. and Brooks J. J. *Creep of Plain and Structural Concrete*. Construction Press, London, UK, 1983.
20. Harvey R. J. and Lenczer D. Creep prestress losses in concrete masonry. *Proceedings of 5th RILEM International Symposium on Creep and Shrinkage in Concrete, Barcelona*, 1993, 71–76
21. Nordström T. and Svensson B. Using and designing massively parallel computers for artificial neural networks. *Journal of Parallel and Distributed Computing*, 1992, **14**, No. 3, 260–285.
22. Bishop C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1996.
23. Ripley B. D. *Pattern Recognition and Neural Networks*. Cambridge University Press, New York, 1996.
24. Tsoukalas L. H. and Uhrig R. E. *Fuzzy and Neural Approaches in Engineering*. Wiley, New York, 1997.
25. Jang J., Sun C. and Mizutani E. *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, Englewood Cliffs, NJ, 1997.
26. Ooyen A. V. and Nienhuis B. Improving the convergence of the back-propagation algorithm. *Neural Networks*, 1992, **5**, 465–471.
27. Haykin S. *Neural Networks: A Comprehensive Foundation*. 2nd Edition, Prentice Hall, Englewood Cliffs, NJ, 1999.
28. Shrive N. G. and Tilleman D. Creep tests on clay masonry

prisms: apparatus and some initial results. *Proceedings of the 7th Canadian Masonry Symposium, Hamilton*, 1995, 1128–1139.

29. Demuth H. and Beale M. *Neural Network Toolbox for Use with MATLAB®*. The Mathworks Inc., MA, 2001.

30. Gentle J. E. *Computational Statistics*. Springer Verlag, New York, 2002.

31. Martinez W. L. and Martinez A. R. *Computational Statistics Handbook with MATLAB®*. Chapman & Hall/CRC Press, New York, 2002.

Please email, fax or post your discussion contributions to the secretary by 1 February 2005: email: daniela.wong@ice.org.uk; fax: +44 (0)20 7665 2249; or post to Daniela Wong, Journals Department, Institution of Civil Engineers, 1–7 Great George Street, London SW1P 3AA.