# Load-Settlement Modeling of Axially Loaded Drilled Shafts using CPT-Based Recurrent Neural Networks

**Mohamed A. Shahin**

Associate Professor, Department of Civil Engineering, Curtin University, Australia

(m.shahin@curtin.edu.au)

**Abstract:** The design of pile foundations requires good estimation of the pile load-carrying capacity and settlement. Design for bearing capacity and design for settlement have been traditionally carried out separately. However, soil resistance and settlement are influenced by each other and the design of pile foundations should thus consider the bearing capacity and settlement in-separately. This requires the full load-settlement response of piles to be well predicted. However, it is well known that the actual load-settlement response of pile foundations can only be obtained by load tests carried out in-situ, which are expensive and time-consuming. In this technical note, recurrent neural networks (RNNs) were used to develop a prediction model that can resemble the full load-settlement response of drilled shafts (bored piles) subjected to axial loading. The developed RNN model was calibrated and validated using several in-situ full-scale pile load tests, as well as cone penetration test (CPT) data. The results indicate that the RNN model has the ability to reliably predict the load-settlement response of axially loaded drilled shafts and can thus be used by geotechnical engineers for routine design practice.

**Keywords:** Drilled shafts; Bored piles; Load-settlement; Modeling; Recurrent neural networks.

## Introduction

Bearing capacity and settlement are the two main criteria that govern the design process of pile foundations so that safety and serviceability requirements are achieved. Design for bearing capacity is carried out by determining the allowable pile load, which is obtained by dividing the ultimate pile load by an assumed factor of safety. Design for settlement, on the other hand, consists of obtaining the amount of settlement that occurs when the allowable load is applied to the pile, causing the soil to consolidate or compress. Design for bearing capacity and design for settlement have been traditionally carried out separately. However, Fellenius (1988) stated that: "*The allowable load on the pile should be governed by a combined approach considering soil resistance and settlement inseparately acting together and each influencing the value of the other*". In addition, there is a strong argument regarding the definition of the ultimate pile load and many methods have been proposed in the literature, some result in interpreted ultimate loads that greatly depend on judgement and the shape of the load-settlement curve. Consequently, for design purposes, the full load-settlement response of piles needs to be well predicted and simulated; the designer can thus decide the ultimate load and comply with the serviceability requirement.

Good prediction of the full load-settlement response of pile foundations needs a thorough understanding of the load transfer along the pile length, which is complex, indeterminate and difficult to quantify (Reese et al. 2006). The actual load-settlement response of pile foundations can only be obtained by carrying out load tests in-situ, which is expensive and time-consuming. On the other hand, the load-settlement response of pile foundations can be estimated using many methods available in the literature. However, due to many complexities, these available methods, by necessity, simplify the problem by incorporating several assumptions associated with the factors that affect the pile behaviour. Therefore, most existing methods failed to achieve consistent success in relation to predictions of pile

capacity and corresponding settlement. In this respect, the artificial intelligence techniques such as artificial neural networks (ANNs) will be efficient as they can resemble the in-situ full-scale pile load tests without the need for any assumptions or simplifications. ANNs are a data mining statistical approach that has proved its potential in many applications in geotechnical engineering and interested readers are referred to Shahin et al. (2001), where the pre-2001 applications are reviewed in some detail, and Shahin et al. (2009) and Shahin (2013), where the post-2001 applications are briefly examined or acknowledged. In recent years, ANNs have been used with varying degrees of success for prediction of axial and lateral bearing capacities of pile foundations in compression and uplift, including driven piles ( Chan et al. 1995; Goh 1996; Lee and Lee 1996; Teh et al. 1997; Abu-Kiefa 1998; Goh et al. 2005; Das and Basudhar 2006; Pal 2006; Shahin and Jaksa 2006; Ahmad et al. 2007; Ardalan et al. 2009; Shahin 2010; Alkroosh and Nikraz 2011) and drilled shafts (Goh et al. 2005; Shahin 2010; Alkroosh and Nikraz 2011). However, to the author's best knowledge, ANNs have not been previously used for modelling load-settlement response of pile foundations and this technical note will fill in part of this gap.

In this paper, the feasibility of using one of ANN techniques, i.e. recurrent neural networks (RNNs), is investigated for modelling load-settlement response of drilled shafts (or bored piles) subjected to axial loading. As mentioned by Briaud et al. (1986), the problem of piles all in sand or all in clay seems to be handled reasonably well by many methods; however, the difficulty arises when the piles are embedded through layered soils. Moreover, Bovolenta (2003) indicated that the design of large diameter bored piles, with particular attention to those embedded in sand, is quite complex because of the influence of many factors that are not simple to estimate or take into account. In the current work, the RNN model is developed for any soil type including layered soils and is valid for bored piles up to 1798 mm in diameter. To facilitate the use of developed RNN model for routine use by

practitioners, the model was translated into an executable program that is made available for interested readers upon request.

## Overview of recurrent neural networks

The type of artificial neural networks used in this study are multilayer perceptrons (MLPs) that are trained with the back-propagation algorithm (Rumelhart et al. 1986). A comprehensive description of backpropagation MLPs is beyond the scope of this technical note but can be found in Fausett (1994). The typical MLP consists of a number of processing elements or nodes that are arranged in layers: an input layer; an output layer; and one or more intermediate layers called hidden layers. Each processing element in a specific layer is linked to the processing element of the other layers via weighted connections. The input from each processing element in the previous layer is multiplied by an adjustable connection weight. The weighted inputs are summed at each processing element, and a threshold value (or bias) is either added or subtracted. The combined input is then passed through a nonlinear transfer function (e.g. sigmoidal or tanh function) to produce the output of the processing element. The output of one processing element provides the input to the processing elements in the next layer. The propagation of information in MLPs starts at the input layer, where the network is presented with a pattern of measured input data and the corresponding measured outputs. The outputs of the network are compared with the measured outputs, and an error is calculated. This error is used with a learning rule to adjust the connection weights to minimize the prediction error. The above procedure is repeated with presentation of new input and output data until some stopping criterion is met. Using the above procedure, the network can obtain a set of weights that produces input-output mapping with the smallest possible error. This process is called “training” or “learning”, which once has been

successful, the performance of the trained model has to be verified using an independent validation set.

In simulations of the typical non-linear response of pile load-settlement curves, the current state of load and settlement governs the next state of load and settlement; thus, recurrent neural networks (RNNs) are recommended. The RNNs proposed by Jordan (1986) imply an extension of the MLPs with current-state units, which are processing elements that remember past activity (i.e. memory units). RNNs then have two sets of input neurons: plan units and current-state units (see Fig. 1). At the beginning of the training process, the first pattern of input data is presented to the plan units while the current-state units are set to zero. As mentioned earlier, the training proceeds, and the first output pattern of the network is produced. This output is copied back to the current-state units for the next input pattern of data. The RNN model development for load-settlement response of drilled shafts is described in detail below.

## Development of RNN model

In this work, the RNN model was developed with the computer-based software package Neuroshell 2, Release 4.2 (Ward 2007). The data used to calibrate and validate the model were obtained from the literature and included a series of 38 in-situ full-scale pile load-settlement tests reported by Alsaman (1995). The tests were conducted on sites of different soil types and geotechnical conditions, ranging from cohesive clays to cohesionless sands including layered soils. The pile load tests include compression loading conducted on straight and belled concrete shafts, and the drilled shafts used had stem diameters ranging from 305 to 1798 mm and embedded lengths from 6 to 27.4 m.

### *Model inputs and outputs*

Seven factors affecting the capacity of drilled shafts were presented to the plan units of the RNN as potential model input variables (Fig. 2). These include the pile stem diameter, *D*, ratio of the pile belled diameter to stem diameter, *B*/*D*, pile embedment length, *L*, weighted average cone point resistance over pile tip failure zone, $\overline{q}_{c-tip}$, weighted average friction ratio over pile tip failure zone, $\overline{f}_{R-tip}$, weighted average cone point resistance over pile embedment length, $\overline{q}_{c-shaft}$, and weighted average friction ratio over pile embedment length, $\overline{f}_{R-shaft}$. The friction ratio, $f_R$, is the ratio of the cone point resistance, $q_c$, to the cone sleeve friction, $f_s$, i.e. $f_R = f_s / q_c$. The current state units of the RNN were represented by three input variables including the axial strain, $\varepsilon_{a,i}$, (= pile settlement/pile diameter), axial strain increment, $\Delta\varepsilon_{a,i}$, and pile load, $Q_i$. The single model output variable is the pile load at the next state of loading, $Q_{i+1}$.

In this study, an axial strain increment that increases by 0.05% was used, in which $\Delta\varepsilon_{a,i}$ = (0.1, 0.15, 0.2, …, 1.0, 1.05, 1.1, …) were utilized. As recommended by Penumadu and Zhao (1999), using varying strain increment values results in good modelling capability without the need for a large size training data. Because the data points needed for RNN model development were not recorded at the above strain increments in the original pile load-settlement tests, the load-settlement curves were digitized to obtain the required data points. This was carried out using Microcal Origin Version 6.0 (Microcal 1999) and implementing linear interpolation. A range between 14 to 28 training patterns was used to represent a single pile load-settlement test, depending on the maximum strain values available for each test. It should be noted that the following conditions were applied to the input and output variables used in the RNN model:

- The pile tip failure zone over which $\bar{q}_{c-tip}$ and $\bar{f}_{R-tip}$ were calculated is taken in accordance with Alsamman (1995), equal to one diameter beneath the drilled shaft base.
- Both the values of cone point resistance and friction ratio are incorporated as model inputs, allowing the soil type (classification) to be implicitly considered in the RNN model. However, it worthwhile noting that records of cone sleeve friction, $f_s$, used to calculate the friction ratio, $f_R$, were not available in the database used; thus, the values of $f_s$ were calculated from the corresponding values of $q_c$ using the curves provided in Alsamman (1995).
- Several CPT tests used in this work include mechanical rather than electric CPT data, thus it was necessary to convert the mechanical CPT readings into equivalent electric CPT values as the electric CPT is the one that is commonly used at present. This is carried out for the cone point resistance using the following correlation proposed by Kulhawy and Mayne (1990):

$$\left(\frac{q_c}{p_a}\right)_{Electric} = 0.47\left(\frac{q_c}{p_a}\right)^{1.19}_{Mechanical} \quad (1)$$

- For the cone sleeve friction, the mechanical cone gives higher reading than the electric cone in all soils with a ratio in sands of about 2, and 2.5–3.5 for clays (Kulhawy and Mayne 1990). In the current work, a ratio of 2 was used for sands and 3 for clays.

### *Data division and preprocessing*

The next step in development of the RNN model is dividing the available data into their subsets. In this work, the data were randomly divided into two sets: a training set for model calibration and an independent validation set for model verification. In total, 34 in-situ pile

load tests were used for model training and 4 tests for model validation. A summary of the data used in the training and validation sets as well as the minimum values, maximum values, ranges and averages, is given in Table 2. Once the available data are divided into their subsets, the input and output variables are preprocessed; in this step the variables were scaled between 0.0 and 1.0 to eliminate their dimensions and ensure that all variables receive equal attention during training.

### *Network architecture and optimization*

Following the data division and the preprocessing, the optimum model architecture (i.e., the number of hidden layers and corresponding number of hidden nodes) must be determined. It should be noted that a network with one hidden layer can approximate any continuous function if sufficient connection weights are used (Hornik et al. 1989). Therefore, one hidden layer was used in the current study. The optimal number of hidden nodes was obtained by a trial-and-error approach in which the network was trained with a set of random initial weights and a fixed learning rate of 0.1; a momentum term of 0.1; a tanh transfer function in the hidden layer nodes; and a sigmoidal transfer function in the output layer nodes. The following number of hidden layer nodes were then utilized: 2, 4, 6, …, and (2$I$+1), where $I$ is the number of input variables. It should be noted that (2$I$+1) is the upper limit for the number of hidden layer nodes needed to map any continuous function for a network with $I$ inputs, as discussed by Caudill (1988). To obtain the optimum number of hidden layer nodes, it is important to strike a balance between having sufficient free parameters (connection weights) to enable representation of the function to be approximated and not having too many, so as to avoid overtraining (Shahin and Indraratna 2006).

To determine the criterion that should be used to terminate the training process, the normalized mean squared error between the actual and predicted values of all outputs in the

training set over all patterns is monitored until no significant improvement in the error occurs. This was achieved at approximately 10,000 training cycles (epochs). Fig. 3 shows the impact of the number of hidden layer nodes on the performance of the RNN model. It can be seen that the RNN model improves with increasing numbers of hidden layer nodes; however, there is little additional impact on the predictive ability of the model beyond 10 hidden layer nodes. Fig. 3 also shows that the network with 14 hidden layer nodes has the lowest prediction error; however, the network with 10 hidden nodes can be considered optimal: its prediction error is not far from that of the network with 14 hidden nodes, and it has fewer connection weights and is thus less complex. As a result of training, the optimal network produced 10 × 10 weights and 10 bias values connecting the input layer to the hidden layer and 10 × 1 weights and one bias value connecting the hidden layer to the output layer.

### *Model performance and validation*

The performance of the optimum RNN model in the training and validations sets is given numerically in Table 1. It can be seen that two different standard performance measures are used, including the coefficient of correlation, $r$, the coefficient of determination (or efficiency), $R^2$. The formulas of these two measures are as follows:

$$r = \frac{\sum_{i=1}^{N}(O_i - \bar{O})(P_i - \bar{P})}{\sqrt{\sum_{i=1}^{N}(O_i - \bar{O})^2 \sum_{i=1}^{N}(P_i - \bar{P})^2}} \qquad (2)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(O_i - P_i)^2}{\sum_{i=1}^{N}(O_i - \bar{O})^2} \quad (3)$$

where; $N$ is the number of data points presented to the model; $O_i$ and $P_i$ are the observed and predicted outputs, respectively; and $\bar{O}$ and $\bar{P}$ are the mean of the predicted and observed outputs, respectively.

The coefficient of correlation, $r$, is a measure that is used to determine the relative correlation between the predicted and observed outputs. However, as indicated by Das and Sivakugan (2010), $r$ sometimes may not necessarily indicate better model performance due to the tendency of the model to deviate toward higher or lower values, particularly when the data range is very wide and most of the data are distributed about their mean. Consequently, the coefficient of determination, $R^2$, is used as it can give unbiased estimate and may be a better measure for model performance. The performance measures in Table 1 indicate that the optimum RNN model performs well and has good prediction accuracy in both the training and validation sets. Table 1 also indicates that the RNN model has consistent performance on the validation set with that obtained on the training set.

The performance of the optimum RNN model in the training and testing sets is further investigated graphically, as shown in Figs. 4 and 5. It should be noted that, for brevity, only five of the most appropriate simulation results in the training set are given in Fig. 4. These five simulations are chosen because they reflect the entire range of the in-situ pile load-settlement tests used in this study. As can be seen in Figs. 4 and 5, excellent agreement between the actual pile load tests and the RNN model predictions is obtained, in both the training and validation sets. The nonlinear relationships of the load-settlement response are

well predicted, and the results demonstrate that the RNN model has a strong capability to simulate the behaviour of drilled shafts quite well.

### *Model robustness and sensitivity analyses*

To further examine the generalization ability (or robustness) of the RNN model, sensitivity analyses are carried out that investigate the response of the RNN predicted pile behavior to a set of hypothetical input data that lie on the range of the data used for model training. For example, to investigate the effect of one parameter such as pile diameter, *D*, all other input variables are set to selected constant values, while *D* is allowed to change. The inputs are then accommodated in the RNN model and the predicted pile load versus settlement response is calculated. This process is repeated for the next input variable and so on, until the model response is examined for all inputs. The robustness of the RNN model is determined by examining how well the predictions compare with the available geotechnical knowledge and experimental data.

The results of the sensitivity analyses are shown in Fig. 6 and indicate that the predicted pile behavior by the RNN model is in good agreement with what one would expect and with published experimental results. For example, it can be observed that the ultimate pile capacity increases with the increase of pile diameter, pile embedment length, soil resistance at the pile tip, and soil resistance at the shaft length. The above results confirm the predictive ability of the developed RNN model in reflecting the role of important factors affecting drilled shafts behavior, and this indicates that the model is robust and can thus be used with confidence.

## Conclusion

The work presented in this technical note has used a series of full-scale in-situ pile load-settlement tests and CPT data collected from the literature to develop a recurrent neural network (RNN) model for simulating the load-settlement response of drilled shafts (or bored piles). The graphical comparison of the load-settlement curves between the RNN model and experiments showed an excellent agreement and indicates that the RNN model can capture the highly non-linear load-settlement response of drilled shafts reasonably well. To facilitate the use of the developed RNN model, it is translated into an executable program using MATLAB and is made available for interested readers upon request.

It is worthwhile noting that predictions from ANN models are better when used for ranges of input variables similar to those utilized in model training. Consequently, the developed RNN model performs best when it is used for the ranges of inputs shown in Table 2. However, these ranges accommodate those values that are usually used in practice and the developed RNN model has the advantage that it can always be updated in the future by presenting new training examples of wider ranges, when new data become available.

## References


Abu-Kiefa, M. A. (1998). "General regression neural networks for driven piles in cohesionless soils." *Journal of Geotechnical & Geoenvironmental Engineering*, 124(12), 1177-1185.

Ahmad, I., El Naggar, H., and Kahn, A. N. (2007). "Artificial neural network application to estimate kinematic soil pile interaction response parameters." *Soil Dynamics and Earthquake Engineering*, 27(9), 892-905.

Alkroosh, I., and Nikraz, H. (2011). "Simulating pile load-settlement behavior from CPT data using intelligent computing." *Central European Journal of Engineering*, 1(3), 295-305.

Alsamman, O. M. (1995). "The use of CPT for calculating axial capacity of drilled shafts," PhD Thesis, University of Illinois-Champaign, Urbana, Illinois.

Ardalan, H., Eslami, A., and Nariman-Zadeh, N. (2009). "Piles shaft capacity from CPT and CPTU data by polynomial neural networks and genetic algorithms." *Computers and Geotechnics*, 36(4), 616-625.

Bovolenta, R. (2003). "Soil stiffness non-linearity in the settlement analysis of foundations," PhD Thesis, University of Genoa, Genoa, Italy.

Briaud, J. L., Tucker, L. M., Anderson, J. S., Perdomo, D., and Coyle, H. M. (1986). "Development of an improved pile design procedure for single piles in clays and sands." *Research Report 4981-1*, Civil Engineering, Texas A&M University, Texas.

Caudill, M. (1988). "Neural networks primer, Part III." *AI Expert*, 3(6), 53-59.

Chan, W. T., Chow, Y. K., and Liu, L. F. (1995). "Neural network: An alternative to pile driving formulas." *Computers and Geotechnics*, 17, 135-156.

Das, S. K., and Basudhar, P. K. (2006). "Undrained lateral load capacity of piles in clay using artificial neural network." *Computers and Geotechnics*, 33(8), 454-459.

Das, S. K., and Sivakugan, N. (2010). "Discussion of: intelligent computing for modeling axial capacity of pile foundations." *Canadian Geotechnical Journal*, 47, 928-930.

Fausett, L. V. (1994). *Fundamentals neural networks: Architecture, algorithms, and applications*, Prentice-Hall, Englewood Cliffs, New Jersey.

Fellenius, B. H. (1988). "Unified design of piles and pile groups." *Transportation Research Record*, 1169, 75-81.

336 Goh, A. T., Kulhawy, F. H., and Chua, C. G. (2005). "Bayesian neural network analysis of
337 undrained side resistance of drilled shafts." *Journal of Geotechnical and*
338 *Geoenvironmental Engineering*, 131(1), 84-93.
339 Goh, A. T. C. (1996). "Pile driving records reanalyzed using neural networks." *Journal of*
340 *Geotechnical Engineering*, 122(6), 492-495.
341 Hornik, K., Stinchcombe, M., and White, H. (1989). "Multilayer feedforward networks are
342 universal approximators." *Neural Networks*, 2, 359-366.
343 Jordan, M. I. (1986) "Attractor dynamics and parallelism in a connectionist sequential
344 machine." *Proceedings of the 8th Annual Conference of the Cognitive science Society*,
345 Amherst, MA, 531-546.
346 Kulhawy, F. H., and Mayne, P. W. (1990). "Manual on estimating soil properties for
347 foundation design " *Report EL-6800*, Electric Power Research Institute, Palo Alto,
348 CA.
349 Lee, I. M., and Lee, J. H. (1996). "Prediction of pile bearing capacity using artificial neural
350 networks." *Computers and Geotechnics*, 18(3), 189-200.
351 Microcal. (1999). *Microcal Origin Version 6.0*, Microcal Software, Inc., Northampton, MA.
352 Pal, M. (2006). "Support vector machine-based modelling of seismic liquefaction potential."
353 *International Journal for Numerical and Analytical Methods in Geomechanics*,
354 30(10), 983-996.
355 Penumadu, D., and Zhao, R. (1999). "Triaxial compression behavior of sand and gravel using
356 artificial neural networks (ANN)." *Computers and Geotechnics*, 24(3), 207-230.
357 Reese, L. C., Isenhower, W. M., and Wang, S. T. (2006). *Analysis and design of shallow and*
358 *deep foundations*, John Wiley & Sons, New Jersey.
359 Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning internal representation
360 by error propagation." Parallel Distributed Processing, D. E. Rumelhart and J. L.
361 McClelland, eds., MIT Press, Cambridge.
362 Shahin, M. A. (2010). "Intelligent computing for modelling axial capacity of pile
363 foundations." *Canadian Geotechnical Journal*, 47(2), 230-243.
364 Shahin, M. A. (2013). "Artificial intelligence in geotechnical engineering: applications,
365 modeling aspects, and future directions." Metaheuristics in Water, Geotechnical and
366 Transport Engineering, X. Yang, A. H. Gandomi, S. Talatahari, and A. H. Alavi, eds.,
367 Elsevier Inc., London, 169-204.

Shahin, M. A., and Indraratna, B. (2006). "Modelling the mechanical behaviour of railway ballast using artificial neural networks." *Canadian Geotechnical Journal*, 43(1), 1144-1152.

Shahin, M. A., and Jaksa, M. B. (2006). "Pullout capacity of small ground anchors by direct cone penetration test methods and neural networks." *Canadian Geotechnical Journal*, 43(6), 626-637.

Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2001). "Artificial neural network applications in geotechnical engineering." *Australian Geomechanics*, 36(1), 49-62.

Shahin, M. A., Jaksa, M. B., and Maier, H. R. (2009). "Recent advances and future challenges for artificial neural systems in geotechnical engineering applications." *Journal of Advances in Artificial Neural Systems*, doi: 10.1155/2009/308239.

Teh, C. I., Wong, K. S., Goh, A. T. C., and Jaritngam, S. (1997). "Prediction of pile capacity using neural networks." *Journal of Computing in Civil Engineering*, 11(2), 129-138.

Ward. (2007). *NeuroShell 2 Release 4.2*, Ward Systems Group, Inc., Mass.

Table
Click here to download Table: Tables.docx

**Table 1.** Performance results of optimal RNN model

| Data sets | Performance measures | |
|---|---|---|
| | $r$ | $R^2$ |
| Training | 0.996 | 0.984 |
| Validation | 0.969 | 0.933 |

**Table 2.** Summary of data used for development of the RNN model

| Test No | $D$ (mm) | $B/D$ | $L$ (m) | $q_{c\text{-}tip}$ (MPa) | $f_{R\text{-}tip}$ (%) | $q_{c\text{-}shaft}$ (MPa) | $f_{R\text{-}shaft}$ (%) | RNN status[a] |
|---|---|---|---|---|---|---|---|---|
| 1 | 305 | 1.00 | 11.5 | 7.0 | 1.23 | 3.4 | 2.35 | T |
| 2 | 838 | 1.00 | 27.4 | 13.3 | 0.65 | 10.6 | 0.81 | T |
| 3 | 1798 | 1.00 | 10.0 | 7.0 | 1.23 | 4.2 | 2.04 | T |
| 4 | 838 | 1.00 | 24.4 | 47.5 | 0.20 | 9.1 | 0.68 | T |
| 5 | 671 | 1.00 | 10.2 | 21.5 | 0.62 | 28.8 | 0.46 | T |
| 6 | 399 | 2.25 | 9.5 | 24.4 | 0.39 | 14.5 | 0.58 | T |
| 7 | 1079 | 1.00 | 13.0 | 39.5 | 0.34 | 21.5 | 0.60 | T |
| 8 | 357 | 1.00 | 10.7 | 8.2 | 1.05 | 3.0 | 2.41 | T |
| 9 | 762 | 1.00 | 13.7 | 4.7 | 1.84 | 3.5 | 2.33 | T |
| 10 | 457 | 1.00 | 15.2 | 1.9 | 2.63 | 8.6 | 0.52 | T |
| 11 | 357 | 1.00 | 10.6 | 8.2 | 1.05 | 5.3 | 1.62 | T |
| 12 | 762 | 1.00 | 16.8 | 5.7 | 1.29 | 5.2 | 1.42 | T |
| 13 | 500 | 1.00 | 10.2 | 14.6 | 0.79 | 3.6 | 1.53 | T |
| 14 | 1091 | 1.00 | 27.0 | 24.4 | 0.35 | 15.3 | 0.47 | T |
| 15 | 399 | 2.25 | 6.7 | 27.3 | 0.49 | 4.8 | 2.00 | T |
| 16 | 1100 | 1.00 | 13.0 | 15.3 | 0.56 | 5.2 | 1.39 | T |
| 17 | 1100 | 1.39 | 6.0 | 17.3 | 0.50 | 17.1 | 0.50 | T |
| 18 | 357 | 1.00 | 12.5 | 7.6 | 1.14 | 3.2 | 2.42 | T |
| 19 | 600 | 1.00 | 12.0 | 20.8 | 0.46 | 12.3 | 0.68 | T |
| 20 | 1298 | 1.46 | 13.5 | 7.6 | 1.01 | 6.9 | 0.83 | T |
| 21 | 521 | 1.00 | 8.2 | 31.8 | 0.42 | 10.3 | 1.12 | T |
| 22 | 762 | 1.00 | 7.0 | 4.7 | 1.84 | 2.2 | 2.43 | T |
| 23 | 1500 | 1.00 | 6.0 | 17.3 | 0.50 | 9.8 | 0.54 | T |
| 24 | 1070 | 1.00 | 25.1 | 24.4 | 0.39 | 9.5 | 0.76 | T |
| 25 | 399 | 2.50 | 11.0 | 24.4 | 0.39 | 17.5 | 0.51 | T |
| 26 | 399 | 2.25 | 8.2 | 24.4 | 0.55 | 3.4 | 2.11 | T |
| 27 | 600 | 1.00 | 7.2 | 15.3 | 0.56 | 6.9 | 0.77 | T |
| 28 | 320 | 2.50 | 6.2 | 25.9 | 0.37 | 13.3 | 0.63 | T |
| 29 | 878 | 1.00 | 10.2 | 7.0 | 1.23 | 4.0 | 2.14 | T |
| 30 | 521 | 1.00 | 8.2 | 31.8 | 0.42 | 10.3 | 1.12 | T |
| 31 | 1100 | 1.00 | 6.0 | 21.5 | 0.42 | 8.5 | 0.90 | T |
| 32 | 814 | 1.00 | 24.2 | 9.6 | 0.90 | 10.5 | 0.73 | T |
| 33 | 399 | 2.13 | 6.5 | 17.3 | 0.50 | 9.0 | 0.85 | T |
| 34 | 320 | 2.03 | 6.6 | 13.3 | 0.61 | 5.5 | 1.34 | T |
| 35 | 457 | 1.00 | 15.2 | 1.9 | 2.63 | 8.6 | 0.52 | V |
| 36 | 600 | 1.00 | 12.0 | 20.8 | 0.46 | 12.3 | 0.68 | V |
| 37 | 320 | 2.50 | 6.2 | 24.4 | 0.39 | 10.7 | 0.72 | V |
| 38 | 1298 | 1.00 | 13.0 | 7.2 | 1.19 | 5.0 | 1.72 | V |
| Minimum | 305 | 1.00 | 6.0 | 1.9 | 0.20 | 2.2 | 0.46 | □ |
| Maximum | 1798 | 2.50 | 27.4 | 47.5 | 2.63 | 28.8 | 2.43 | □ |
| Range | 1493 | 1.50 | 21.4 | 45.6 | 2.43 | 26.6 | 1.97 | □ |
| Average | 725 | 1.30 | 12.2 | 17.4 | 0.80 | 9.0 | 1.20 | □ |

[a]T, training; V, validation.

**Figure captions:**

**Fig. 1.** Schematic diagram of recurrent neural networks

**Fig. 2.** Architecture of the developed RNN model

**Fig. 3.** Effect of number of hidden nodes on RNN model performance

**Fig. 4.** Some simulation results of RNN model in the training set

**Fig. 5.** Simulation results of RNN model in the validation set

**Fig. 6.** Sensitivity analyses to test the robustness of RNN model

Fig. 1.

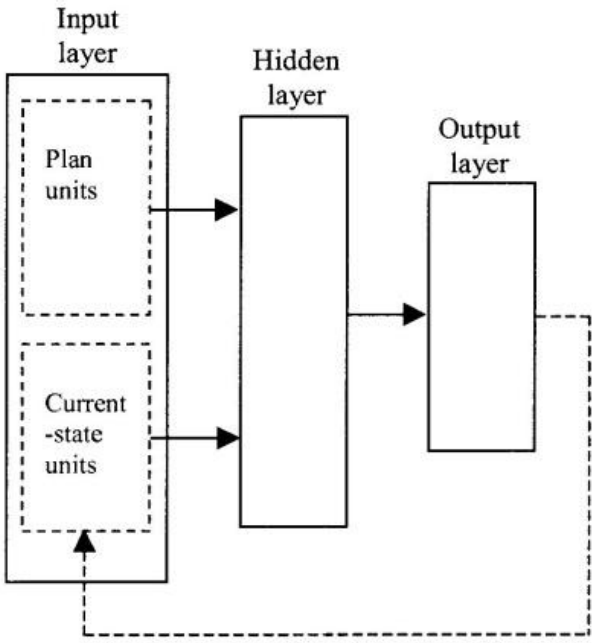

Input
layer
Plan
units
Current
-state
units
Hidden
layer
Output
layer

Fig. 2.

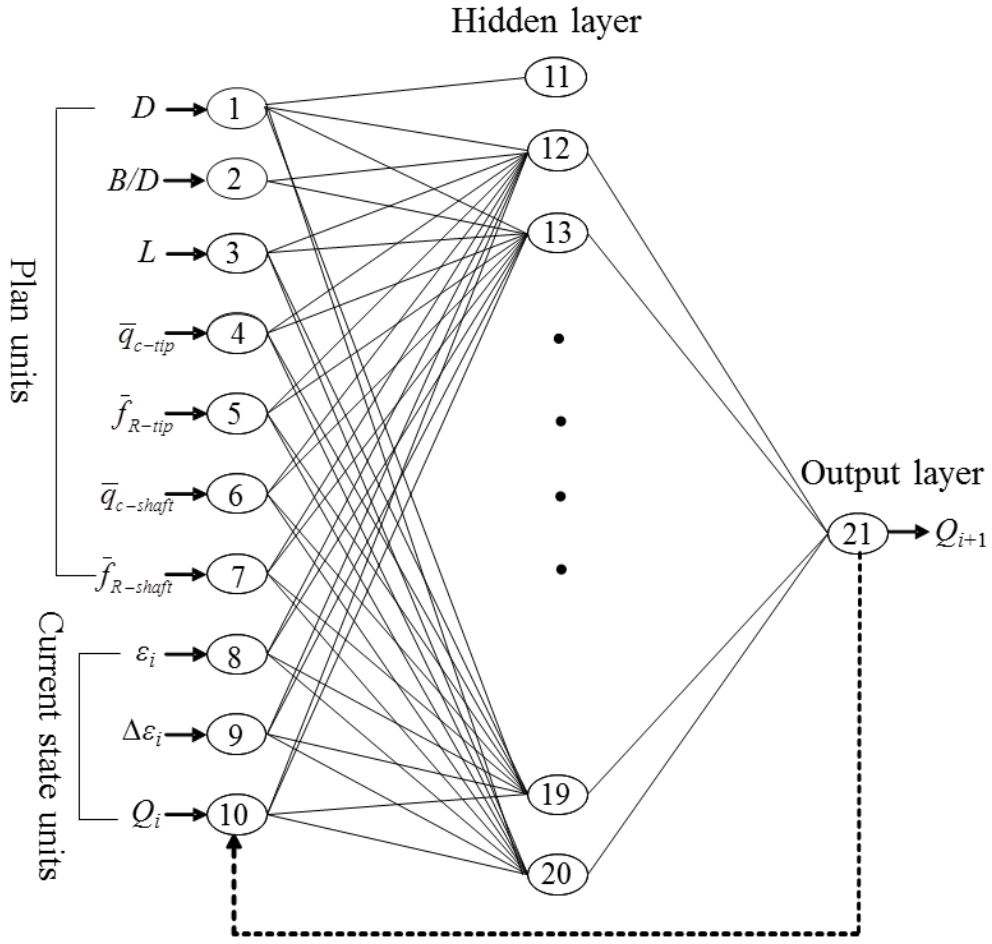

Hidden layer
Plan units
Current state units
Output layer
D
B/D
L
$\bar{q}_{c-tip}$
$\bar{f}_{R-tip}$
$\bar{q}_{c-shaft}$
$\bar{f}_{R-shaft}$
$\varepsilon_i$
$\Delta\varepsilon_i$
$Q_i$
1
2
3
4
5
6
7
8
9
10
11
12
13
19
20
21
$Q_{i+1}$

Fig. 3.

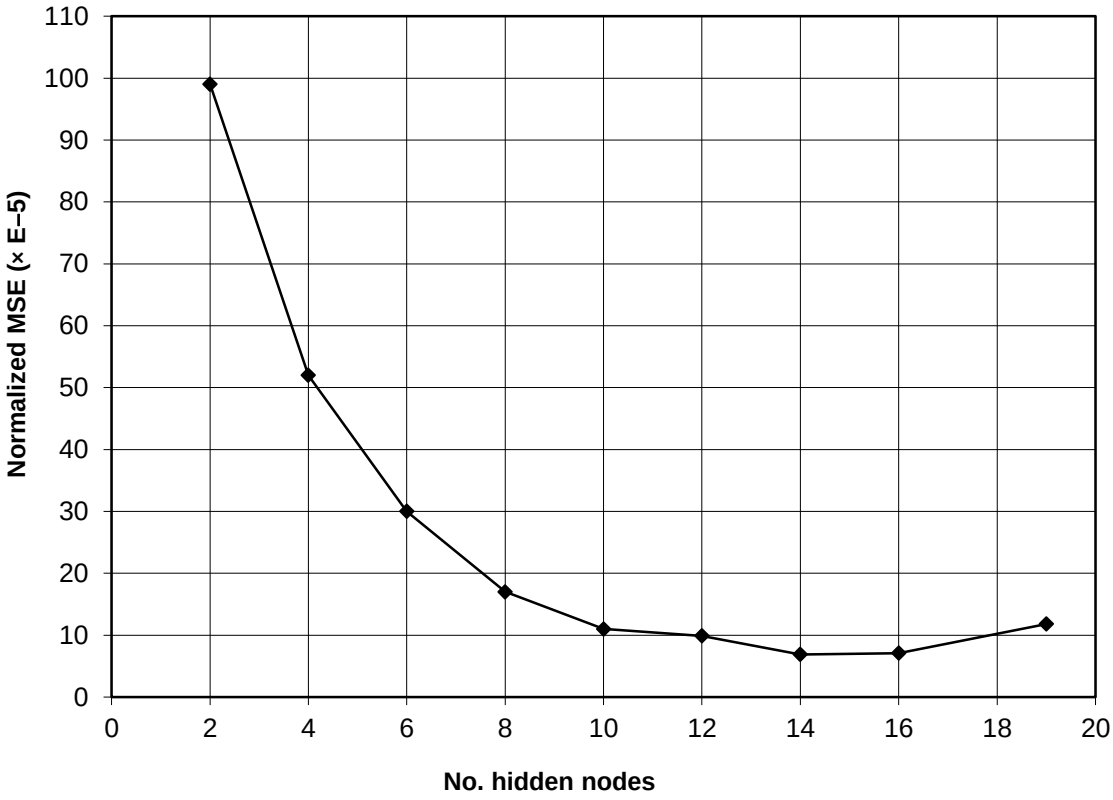

Normalized MSE (× E–5)
110
100
90
80
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
14
16
18
20
No. hidden nodes

Fig. 4.

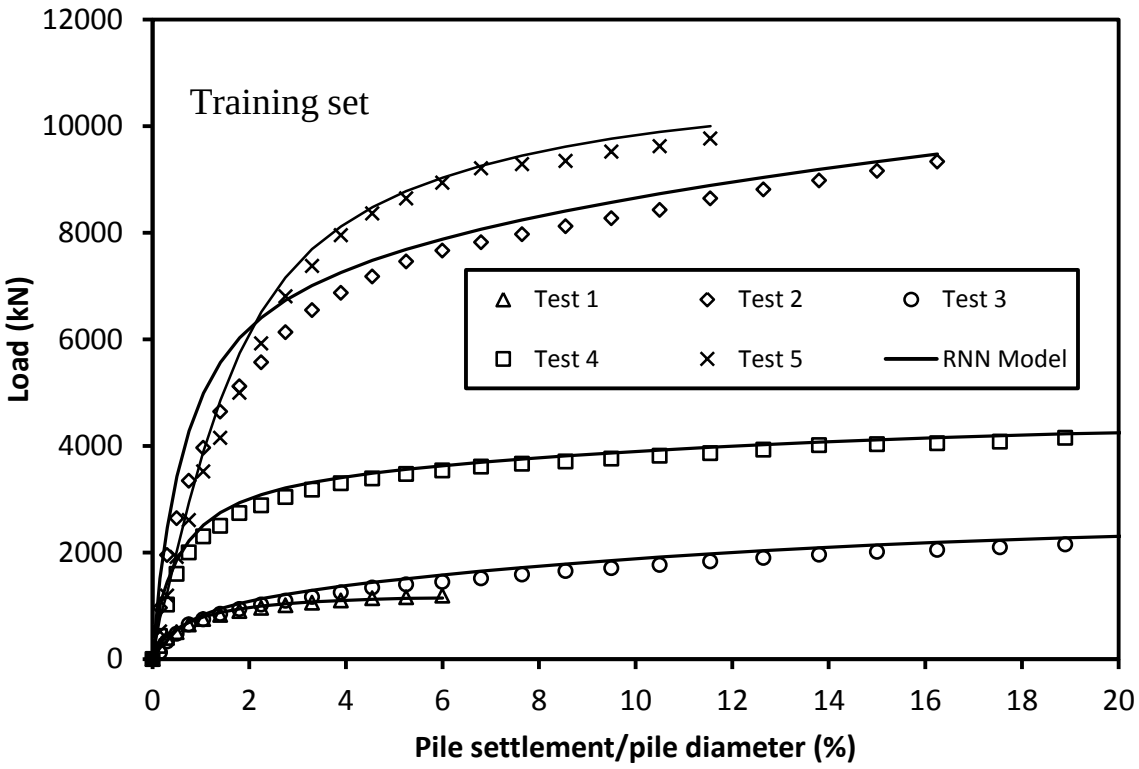

Training set
Load (kN)
Pile settlement/pile diameter (%)
Test 1
Test 2
Test 3
Test 4
Test 5
RNN Model
0
2000
4000
6000
8000
10000
12000
0
2
4
6
8
10
12
14
16
18
20

Fig. 5.

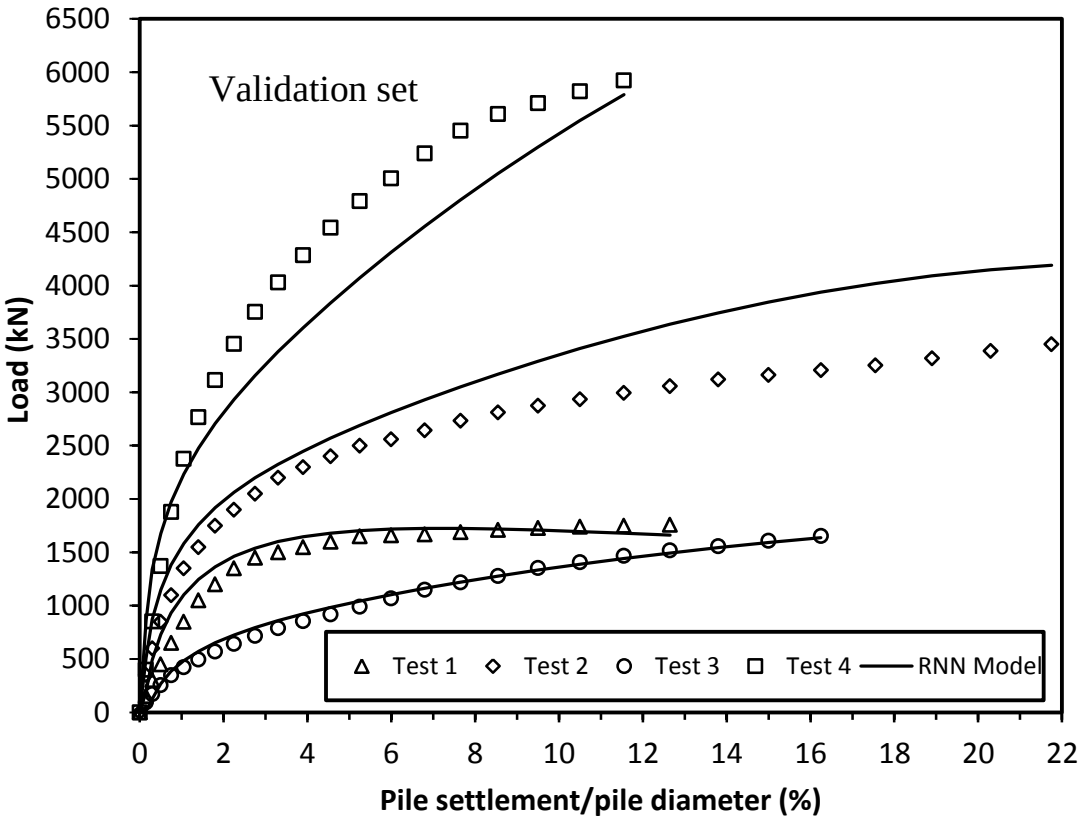
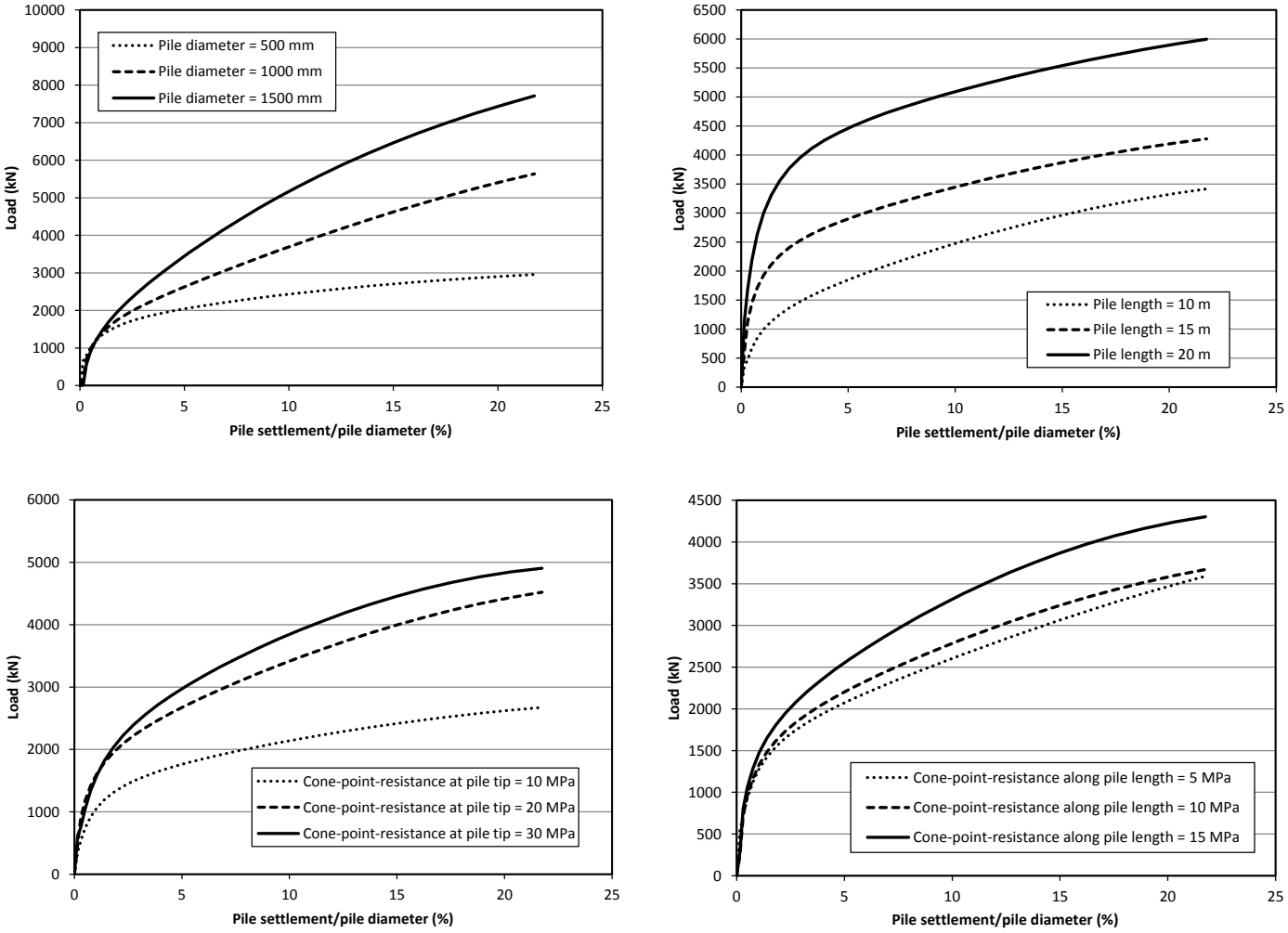

Validation set
Load (kN)
Pile settlement/pile diameter (%)
Test 1
Test 2
Test 3
Test 4
RNN Model
6500
6000
5500
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20
22

Fig. 6.


Load (kN)
Pile settlement/pile diameter (%)
Pile diameter = 500 mm
Pile diameter = 1000 mm
Pile diameter = 1500 mm
Pile length = 10 m
Pile length = 15 m
Pile length = 20 m
Cone-point-resistance at pile tip = 10 MPa
Cone-point-resistance at pile tip = 20 MPa
Cone-point-resistance at pile tip = 30 MPa
Cone-point-resistance along pile length = 5 MPa
Cone-point-resistance along pile length = 10 MPa
Cone-point-resistance along pile length = 15 MPa