Neural and Neurofuzzy Techniques Applied to Modelling Settlement of Shallow Foundations on Granular Soils

M. A. Shahin[a], **H. R. Maier**[b] **and M. B. Jaksa**[b]
[a] Research Associate, School of Civil & Environmental Engineering, University of Adelaide, SA, Australia
Email: mshahin@civeng.adelaide.edu.au
[b] Senior Lecturer, School of Civil & Environmental Engineering, University of Adelaide, SA, Australia

Abstract: This paper describes two modelling techniques applied to a case study of settlement prediction of shallow foundations on granular soils. The first technique uses multi-layer perceptrons (MLPs) that are trained with the back-propagation algorithm, whereas the second technique uses B-spline neurofuzzy networks that are trained with the adaptive spline modelling of observation data (ASMOD) algorithm. The performance of the models obtained using both techniques is assessed in terms of prediction accuracy, model parsimony and model transparency. The results indicate that both the back-propagation MLP and the B-spline neurofuzzy models are comparable in terms of prediction accuracy, although the back-propagation MLP model is found to perform slightly better than the B-spline neurofuzzy model. In terms of model parsimony, the B-spline neurofuzzy model is found to be more parsimonious than the back-propagation MLP model. In terms of model transparency, the B-spline neurofuzzy model is found to provide a more explicit interpretation of the relationships between the model inputs and the corresponding outputs.

Keywords: *Neural networks; Neurofuzzy; Modelling; Settlement prediction; Shallow foundations*

1 INTRODUCTION

Artificial neural networks (ANNs) (Fausett 1994) are numerical modelling techniques that are inspired by the functioning of the human brain and nerve system. ANNs have been applied successfully to many aspects of geotechnical engineering. In the majority of these applications, multi-layer perceptrons (MLPs) that are trained with the back-propagation algorithm are used. This can be attributed to the fact that MLPs trained with back-propagation have a high capability of data mapping. However, one shortcoming of such MLPs is that the knowledge acquired in the trained network is stored in its connection weights in a complex manner that is often difficult to interpret. On the other hand, B-spline neurofuzzy networks that are trained with the adaptive spline modelling of observation data (ASMOD) algorithm can perform input/output data mappings, similar to the way MLPs do, with the additional benefit of being able to translate the acquired knowledge into a set of fuzzy rules that describe the model input/output relationships in a more transparent fashion.

In this paper, the two types of ANNs mentioned above are applied to a case study of predicting settlement of shallow foundations on granular soils. The obtained models are compared in terms of prediction accuracy, model parsimony and model transparency.

2 BACK-PROPAGATION MULTI–LAYER PERCEPTRONS

A typical structure of MLPs consists of a number of processing elements (PEs), or nodes, that are usually arranged in layers: an input layer, an output layer and one or more hidden layers (Figure 1).

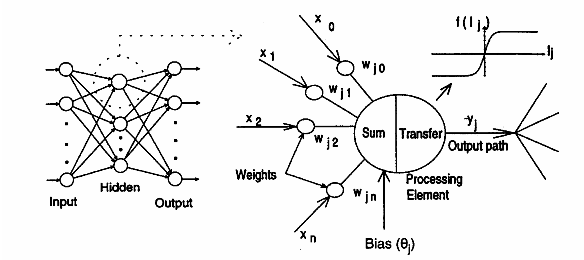

Figure 1. Structure and operation of an MLP

Each PE in a specific layer is fully or partially joined to many other PEs via weighted connections. The input from each PE in the previous layer (x_i) is multiplied by an adjustable connection weight (w_{ji}). At each PE, the weighted input signals are summed and a threshold value or bias (θ_j) is added or subtracted. This combined input (I_j) is then passed through a non-linear transfer function ($f(.)$) (e.g. sigmoidal function or tanh function) to produce the output of the PE (y_j). The output of one PE provides the input to the PEs in the next layer. This process is summarised in Equations 1 and 2 and illustrated in Figure 1.

$$I_j = \sum w_{ji} x_i + \theta_j \qquad \text{summation} \qquad (1)$$

$$y_j = f(I_j) \qquad \text{transfer} \qquad (2)$$

The propagation of information in MLPs starts at the input layer where the network is presented with a historical set of input data. The actual output of the network is compared with the desired output and an error is calculated. Using this error and utilising a learning rule, the network adjusts its weights until some stopping criterion is met so that the network can find a set of weights that will produce the input/output mapping that has the smallest possible error. This process is called “learning” or “training”. One common stopping criterion is the cross-validation technique proposed by Stone (1974). Cross-validation requires the data to be divided into three sets; a training set, a testing set and a validation set. The training set is used to adjust the connection weights, the testing set is used to decide when to stop training to avoid overfitting and the validation set is used to test the predictive ability of the model in real-world situations.

3 B-SPLINE NEUROFUZZY NETWORKS

Neurofuzzy networks use the fuzzy logic system to store the knowledge acquired between a set of input variables (x_1, x_2, …, x_n) and the corresponding output variable (y) in a set of linguistic fuzzy rules that can be easily interpreted, such as: IF (x_1 is *high* AND x_2 is *low*) THEN (y is *high*), c=0.9, where (c=0.9) is the rule confidence which indicates how much the above rule has contributed to the output. As part of any fuzzy logic system, two main components (i.e. fuzzy sets and fuzzy rules) need to be determined. In order to determine the fuzzy sets, linguistic terms (e.g. small, medium and large) can be interpreted mathematically in the form of membership functions, and model variables are *fuzzified* to be partial members of these membership functions in the interval grade (0,1). This means that, for a fuzzy set A, an input variable x is fuzzified to be a partial member of the fuzzy set A by transforming it into a degree of membership of function $u_A(x)$ of interval (0,1). B-spline basis functions are piecewise polynomials of order k that can be used as one form of membership function. For each variable, the fuzzy sets overlap and cover the necessary range of variation for that variable in a process called *fuzzification*. It should be noted that the model output of a fuzzy set is fuzzy too, and in order to obtain a real-valued output, defuzzification is needed. The mean of maxima and centre of gravity are the most popular defuzzification algorithms (Brown and Harris 1994).

A typical structure of a B-spline neurofuzzy network contains three layers: an input layer; a single hidden layer; and an output layer (Brown and Harris 1994). The input layer normalises the input space in a p-dimensional lattice (Figure 2). Each cell of the lattice represents similar regions of the input space. The hidden layer consists of B-spline basis functions, which are defined on the normalised input space. The size, shape and overlap of the basis functions determine the structure and complexity of the network. The output layer sums the weighted outputs from the basis functions to produce the network output using the following equation:

$$y = \sum_{i=1}^{p} a_i w_i \qquad (3)$$

where y = model output; a_i = output from the pth basis function; and w_i = connection weight associated with a_i. This output is compared with the actual measured output and a connection error is calculated. Using this error and implementing a learning rule, the neurofuzzy network adjusts its weights and determines its fuzzy parameters (i.e. fuzzy sets and rules).

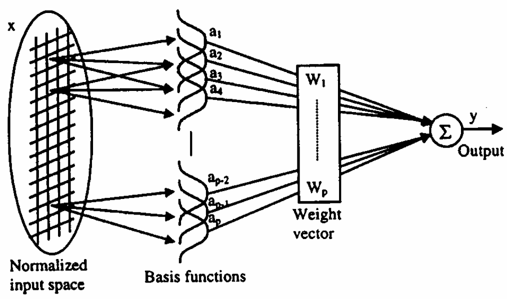

Figure 2. Typical structure of a neurofuzzy network (Brown and Harris 1994).

One major disadvantage of B-spline neurofuzzy networks is the so-called *curse of dimensionality*, in which the number of fuzzy rules is exponentially dependent on the dimension of the input space. This results in a large number of fuzzy rules and, consequently, impractical model representation. The analysis of variance (ANOVA) representation is one useful approach to overcome this problem (Brown and Harris 1994). ANOVA decomposes an n-dimensional function into a linear combination of a number of separate functions, as follows (Brown and Harris 1994):

$$f(x) = f_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{i,j}(x_i, x_j) + \ldots + f_{1,2,\ldots,n}(x) \quad (4)$$

where f_0 represents a constant (the function bias); and the other terms represent the univariate, bivariate and high-order subfunctions. In many situations, the majority of high-order terms are zero or negligible, resulting in a limited number of subfunctions (often called subnetworks) of much lower dimensions that approximate the network input/output mapping. It should be noted that each subnetwork in the ANOVA description represents a neurofuzzy system of its own and the overall model output is produced by summing outputs of all subnetworks.

The adaptive spline modelling of observation data (ASMOD) proposed by Kavli (1993) is an algorithm that can be used to automatically obtain the optimal structure of B-spline neurofuzzy networks and select model inputs that have the most significant impact on outputs. The algorithm starts with a simple model (e.g. only one variable with two membership functions) and iteratively refines the model structure during training so as to gradually increase model capability until some stopping criterion is met. Possible refinements include adding or deleting input variables, forming multi-variate subnetworks using ANOVA, and increasing the number and dimension of an individual subnetwork. For every refinement, the impact of network pruning is evaluated and the network that has the simplest structure with the best performance is chosen. One common stopping criterion is the Bayesian Information Criterion (BIC) given by Brown and Harris (1994), as follows:

$$K = L \ln(MSE) + p \ln(L) \quad (5)$$

where K = performance measure; p = size of current model; MSE = mean square error; and L = number of data pairs used to train the model. The measure, given in Equation 5, balances model complexity, the number of training data, and model error. It should be noted that the BIC stopping criterion requires the data to be divided into two sets; a training set to build the model and an independent validation set to test the predictive ability of the model in real-world situations.

4 CASE STUDY

The case study considered in this research is concerned with predicting the settlement of shallow foundations on granular soils. The settlement of shallow foundations is usually divided into immediate and consolidation settlements. Immediate settlement occurs with load application during, or immediately after, the construction of a structure. It is primarily due to the distortion and reorientation of soil grains. Consolidation settlement, on the other hand, generally takes months to years to occur and is due to the dissipation of pore water pressure over time. For granular soils (sand and gravel) which are the subject of this paper, only the immediate settlement is of interest, whereas consolidation settlement is the major concern for cohesive soils (silt and clay). Immediate settlement of shallow foundations on granular soils usually causes relatively rapid deformations of superstructures, which results in an inability to remedy damage and to avoid further deformation. As a consequence, settlement is a major concern and is an essential criterion in the design process of shallow foundations on granular soils.

It is generally accepted that five parameters have the most significant impact on the settlement of shallow foundations on granular soils (Burland and Burbidge 1985; Shahin et al. 2002b). These include the footing width (B), footing net applied pressure (q), soil compressibility (or density) which can be represented by the average blow count (N) obtained from the standard penetration test (SPT) over the depth of influence of the foundation, footing geometry (L/B) and footing embedment ration (D_f/B). The data used in this research comprise a total of 189 individual cases (Shahin et al. 2002b) that include field measurements of shallow foundations, as well as the corresponding information regarding footings and soils.

As recommended by Burland and Burbidge (1985), the values of N used in this research have not been corrected for overburden pressure nor submergence. However, for very fine and silty sand below the water table, the submergence correction proposed by Terzaghi and Peck (1948), when $N > 15$ is used, as follows:

$$N_{corrected} = 15 + 0.5(N - 15) \quad (6)$$

For gravel or sandy gravel, the correction proposed by Burland and Burbidge (1985) is used, as follows:

$$N_{corrected} = 1.25N \quad (7)$$

It should also be noted that the depth of influence over which N is measured is that proposed by Burland and Burbidge (1985), as follows. When N is decreasing with depth, the depth of influence is taken to be equal to the lesser of $2B$ or the depth from the bottom of the footing to bedrock. On the other hand, when N is constant or increasing with depth, the depth of influence is taken to be equal to $B^{0.75}$.

5 DEVELOPMENT OF ANN MODELS

5.1 Back-propagation MLP Model

The MLP model used in this work was developed by Shahin et al. (2002b) and was implemented using the commercial software package *Neuframe* Version 4.0 (Neusciences 2000). The model inputs were the footing width (B), footing net applied pressure (q), the average SPT blow count (N) over the depth of influence of the foundation, footing geometry (L/B) and footing embedment ration (D_f/B). The measured settlement (S_m) was the single output. The available data were randomly divided into three sets (i.e. training, testing and validation) in such a way that they are statistically consistent and thus represent the same statistical population (Masters 1993). In total, 80% of the data were used for training and 20% were used for validation. The training data were further divided into 70% for the training set and 30% for the testing set. Before presenting the input and output variables for MLP model training, they were scaled between 0.0 and 1.0 to eliminate their dimension and to ensure that all variables receive equal attention during training. The optimal model geometry was determined utilising a trial-and-error approach in which MLP models were trained using one hidden layer with 1, 2, 3, 5, 7, 9 and 11 hidden layer nodes, respectively. It should be noted that one hidden layer can approximate any continuous function, provided that sufficient connection weights are used (Hornik et al. 1989). It should also be noted that 11 is the upper limit for the number of hidden layer nodes needed to map any continuous function for a network with 5 inputs, as proposed by Caudill (1988). The optimal network parameters were obtained by training the MLP model with different combinations of learning rates and momentum terms. A model with 2 hidden layer nodes, a learning rate of 0.2, a momentum term of 0.8, tanh transfer function for the hidden layer nodes and sigmoidal transfer function for the output layer node was found to perform best (Shahin et al. 2002b).

The performance of the MLP model obtained is summarised in Table 1. It can be seen that three different performance measures are used, including the coefficient of correlation (r), the root mean square error (RMSE) and the mean absolute error (MAE). It can also be seen that the model performs well, as it has high coefficients of correlation, r, and low RMSEs and MAEs in the training, testing and validation sets.

Table 1. Performance of the MLP model.

Data set	r	RMSE (mm)	MAE (mm)
Training	0.930	10.01	6.87
Testing	0.929	10.12	6.43
Validation	0.905	11.04	8.78

5.2 B-spline Neurofuzzy Model

The B-spline neurofuzzy network used in this work was developed by Shahin et al. (2003) and was again implemented using the commercial software package *Neuframe* Version 4.0 (Neusciences 2000). The factors affecting settlement, i.e. the footing width (B), footing net applied pressure (q), the average SPT blow count (N) over the depth of influence of the foundation, footing geometry (L/B) and footing embedment ration (D_f/B), are presented to the neurofuzzy model as potential model input variables. The measured settlement (S_m) is the single model output variable. The ASMOD algorithm, described in Section 3, is used for model optimisation. As mentioned previously, the ASMOD algorithm automatically optimises model architecture and selects the input variables that have the most significant impact on model outputs. The ASMOD algorithm also uses stopping criteria (e.g. BIC) that require the data to be divided into two sets; training and validation. In this work, the training and testing sets used to develop the MLP model in Section 5.1 are combined to form the training set for the neurofuzzy network, whereas the validation set is kept the same and thus, a fair comparison between the MLP and neurofuzzy models can be carried out. Using this procedure, 80% of the available data are used for training and 20% are used for validation.

The performance of the neurofuzzy model obtained is shown in Table 2. It can be seen that the model performs well, as it has high coefficients of correlation, r, and low RMSEs and MAEs for the training and validation sets.

Table 2. Performance of the neurofuzzy model.

Data set	*r*	RMSE (mm)	MAE (mm)
Training	0.889	12.33	8.08
Validation	0.881	12.36	9.36

A schematic view of the model obtained is given in Figure 3. It can be seen that the model uses only 3 of the 5 potential input variables as the most significant inputs. The chosen inputs are the footing width (*B*), footing net applied pressure (*q*) and the average SPT blow count (*N*). It can also be seen that the model has one 1D and one 2D subnetwork. In each of the subnetworks obtained, triangular membership functions of order 2 are used for all input variables, as shown in Figure 4. It can be seen from this figure that the membership functions of *B*, *q* and S_m are presented over a two-valued linguistic universe (i.e. small and large for *B*, light and heavy for *q*, and low and high for S_m). On the other hand, the membership functions of the soil density, which is represented herein by the average SPT blow count, *N,* is presented over a four-valued linguistic universe (i.e. loose, medium, dense, and very dense). As a result, the first subnetwork contains 8 rules while the second subnetwork contains 2 rules, resulting in a model with 10 fuzzy rules, as listed in Table 3. It should be noted that the number between brackets in Table 3 represents the rule confidence described in Section 3.

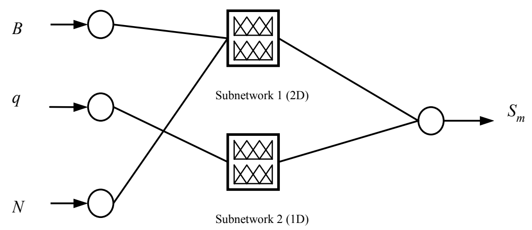

Figure 3. Schematic representation of the neurofuzzy model.

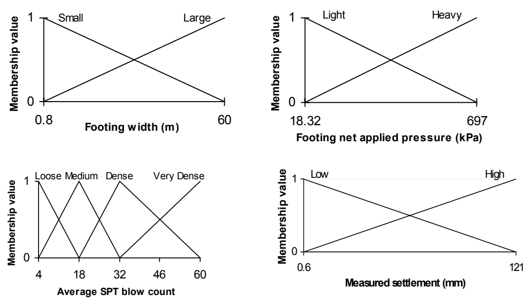

Figure 4. Membership functions of input variables used by the neurofuzzy model.

The fuzzy rules in Table 3 are a valuable source of information from which knowledge can be extracted that governs the relationship between settlement and the factors affecting settlement. It should be noted that the range of applicability of the fuzzy rules in Table 3 is constrained by the quality of the data used in the model calibration phase. Consequently, it is unlikely that they provide a general representation of the relationship between settlement and the factors affecting it. However, in general, the fuzzy rules obtained are in agreement with what one would expect based on the underlying physical meaning of the settlement problem. It can also be seen from Table 3 that Rules 9 and 10 seem to conflict with what one would expect based on the underlying physical meaning of the settlement problem. Rules 9 and 10 indicate that settlement is most likely to be low regardless of whether the applied load is light or heavy. The most likely reason for this is that there were insufficient training data to cover the full range of possible high settlement conditions.

Table 3. Fuzzy rules extracted by the neurofuzzy model.

Subnet work No.	Rule No.	Rule
1	1	IF "Footing width" is *Small* AND "Soil" is *Loose* THEN "Settlement" is *Low* (0.84) OR "Settlement" is *High* (0.16)
	2	IF "Footing width" is *Large* AND "Soil" is *Loose* THEN "Settlement" is *High* (1.00)
	3	IF "Footing width" is *Small* AND "Soil" is *Medium* density THEN "Settlement" is *Low* (0.99) OR "Settlement" is *High* (0.01)
	4	IF "Footing width" is *Large* AND "Soil" is *Medium* density THEN "Settlement" is *Low* (0.44) OR "Settlement" is *High* (0.56)
	5	IF "Footing width" is *Small* AND "Soil" is *Dense* THEN "Settlement" is *Low* (0.96) OR "Settlement" is *High* (0.04)
	6	IF "Footing width" is *Large* AND "Soil" is *Dense* THEN "Settlement" is *Low* (0.86) OR "Settlement" is *High* (0.14)
	7	IF "Footing width" is *Small* AND "Soil" is *Very Dense* THEN "Settlement" is *Low* (1.00)
	8	IF "Footing width" is *Large* AND "Soil" is *Very Dense* THEN "Settlement" is *Low* (0.86) OR "Settlement" is *High* (0.14)
2	9	IF "Net applied pressure" is *Light* THEN "Settlement" is *Low* (0.96) OR "Settlement" is *High* (0.04)
	10	IF "Net applied pressure" is *Heavy* THEN "Settlement" is *Low* (0.87) OR "Settlement" is *High* (0.13)

6 COMPARISON OF MLP AND B-SPLINE NEUROFUZZY MODELS

A comparison between the back-propagation MLP and B-spline neurofuzzy models is carried out in terms of model accuracy, model parsimony and model transparency. A summary of the number of inputs and connection weights used by each model is given in Table 4. The performance results of the two models with respect to the validation set are also given in Table 4. In terms of model accuracy, it can be seen that the two models are comparable, although the MLP model performs slightly better than the neurofuzzy model. In terms of model parsimony, the neurofuzzy model is found to be more parsimonious than the back-propagation MLP model, as it has a smaller number of model inputs and connection weights. In terms of model transparency, the neurofuzzy model is found to provide a more explicit interpretation of the relationships between model inputs and the corresponding output in the form of a set of linguistic fuzzy rules that describe the model in a more transparent fashion (Table 3). However, it was shown by Shahin et al. (2002a) that the small number of hidden layer nodes of the MLP model enabled the translation of the model into a relatively simple equation that provides a valuable insight into the relationships between the model inputs and the corresponding outputs. For large MLP models with a larger number of inputs and hidden layer nodes, a derivation of such an equation could be difficult and consequently, the use of neurofuzzy models would be better in such situations.

Table 4. Comparison between the MLP and neurofuzzy models.

Model type	No. of inputs	No. of connection weights	Model performance on the validation set		
			r	RMSE (mm)	MAE (mm)
MLP	5	12	0.91	11.04	8.78
Neuro-fuzzy	3	8	0.88	12.36	9.36

7 CONCLUSIONS

In this paper, two types of modelling techniques that adopt artificial neural networks (ANNs) were examined for a case study of settlement prediction of shallow foundations on granular soils. The first type was multi-layer perceptrons (MLPs) that are trained with the back-propagation algorithm, whereas the second type was B-spline neurofuzzy networks that are trained with the adaptive spline modelling of observation data (ASMOD) algorithm. The MLP and neurofuzzy models developed were compared in terms of prediction accuracy, model parsimony and model transparency. In terms of prediction accuracy, it was found that the two models are comparable although the MLP model performs slightly better that the neurofuzzy model. In terms of model parsimony, it was found that the neurofuzzy model is more parsimonious than the MLP model with fewer model inputs and connection weights. In terms of model transparency, it was found that the neurofuzzy model is more transparent than the MLP model, as it was able to describe the relationship between model inputs and the corresponding output in the form of a set of fuzzy rules.

8 REFERENCES

Brown, M., and C. Harris, *Neurofuzzy adaptive modelling and control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

Burland, J. B., and M.C. Burbidge, Settlement of foundations on sand and gravel, Proc. Institution of Civil Engineers, London, 78-Part 1, 1325-1381, 1985.

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

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

Hornik, K., M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks*, 2, 359-366, 1989.

Kavli, T., ASMOD – an algorithm for adaptive spline modelling of observation data, *Int. J. Control*, 58(4), 947-967, 1993.

Masters, T., *Practical neural network recipes in C++*, Academic Press, San Diego, California, 1993.

Neusciences, *Neuframe Version 4.0*, Neusciences Corp., Southampton, Hampshire, 2000.

Shahin, M.A., M.B. Jaksa, and H.R. Maier, Artificial neural network-based settlement prediction formula for shallow foundations on granular soils, *Australian Geomechanics*, 37(4), 45-52, 2002a.

Shahin, M.A., M.B. Jaksa, and H.R. Maier, Neurofuzzy networks applied to settlement of shallow foundations on granular soils, *9th Int. Conf. on Applications of Statistics and Probability in Civil Eng.*, University of California, Berkeley, (accepted), 2003.

Shahin, M.A., H.R. Maier, and M.B. Jaksa, Predicting settlement of shallow foundations using neural networks, *J. Geotech. & Geoenv. Eng.*, 128(9), 785-793, 2002b.

Stone, M., Cross-validatory choice and assessment of statistical predictions, *J. Royal Statistical Society*, B 36, 111-147, 1974.

Terzaghi, K., and R.B. Peck, *Soil mechanics in engineering practice*, John Wiley, New York, 1948.