

PREDICTION OF SCOUR DEPTH AROUND BRIDGE PIERS USING EVOLUTIONARY NEURAL NETWORK

ABDUSSAMAD ISMAIL – Senior Lecturer, PhD, Bayero University Kano, Faculty of Engineering,

✉: ismaila2836@buk.edu.ng

Abstract: An empirical formula based on evolutionary regression network is proposed in this paper for predicting the equilibrium depth of scour around bridge piers. The formula expresses the equilibrium scour depth as a function of variables including flow depth and mean velocity, critical flow velocity, median grain size and pier diameter. The empirical formula is developed by training and testing an evolutionary network using scour data available in the literature. The use of the evolutionary algorithm in developing the formula is informed by the need to reduce the model complexity while sacrificing its predictive accuracy. The results of performance comparisons with existing models showed that the proposed formula model produces reasonably accurate estimates of equilibrium scour depth with a much smaller number of fitting constants compared with back-propagation neural networks.

Keywords: Scour depth, Bridge piers, Neural network model

1. Introduction

Design of bridge piers requires an accurate estimate of the worst scour depth around their foundations. The aim is to take adequate preemptive measures to prevent bridge failures caused by local scour around piers and abutments, which over the years caused loss of lives and resulted in damages and replacements worth millions of dollars. In order to provide guidance to practicing engineers, experimental studies have been sponsored by various agencies in order to have a better understanding of the process involved in the development of scour hole around piers. Out of these studies, a number of empirical formulae were developed, which include the models proposed by Laursen and Toch [1], Shen [2], Hancu [3], Breusers et al. [4], USDOT [5], and Melville and Chiew [6]. In Table 1 below, descriptions of some of these approaches are given in summary. The major reason for resorting to empirical correlations for scour depth estimation is the highly complicated nature of fluid-soil –structure interaction involved in the formation scour, which makes it quite challenging to develop an analytical or numerical model that can help in accurately predicting the depth of scour holes.

Despite the adoption of the aforementioned empirical formulae, their inability to adequately simulate the interaction of various factors that contribute to the evolution of scour hole raises a serious concern regarding their reliability. This is evidenced by the large prediction errors recorded when the estimates of the formulae are compared with the independent experimental data [8].

In order to address the prediction difficulties associated with existing empirical methods, the use of machine learning techniques has been explored over the last decade in developing models based on of experimental data accumulated over the years for scour depth predictions. The works of Lee et al [7], Bateni et al [8] and Ismail et al.[9] are part of this effort. Other relevant works include studies by Akib et al. [10], Chou et al. [11], Pang et al. [12] and more recently Ebtehaj et al. [13]. The overall conclusions of these studies are that neural networks successfully managed to produce accurate estimates to scour depths when tested against independent scour data. They were also found significantly surpass the existing scour prediction formulae in terms of accuracy. The success recorded by neural net predictors can be attributed to their ability to explore complex behaviours of systems without having to make apriori assumptions about the how the systems behave. Their key drawback, however, is that they are based on so large a number of constants that end users may not find them convenient to use.

Some of the currently available models for predicting scour depth

Reference	Formula
[1]	$d_{se} = 1.35 D^{0.7} Y^{0.3}$
[3]	$\frac{d_{se}}{D} = 2.42 \left(2 \frac{U}{U_c} - 1 \right) \left(\frac{U_c^2}{gD} \right)^{1/3}$
[5]	$\frac{d_{se}}{D} = 2k_1 k_2 k_3 k_4 \left(\frac{Y}{D} \right)^{0.35} F_r^{0.43}$ where k_1 , k_2 , k_3 and k_4 are, respectively, correction coefficients accounting for pier nose shape, attack angle, bed condition and armouring.
[6]	$d_{se} = k_{yD} k_l k_d$ where k_{yD} = flow depth coefficient; k_l = pier width coefficient and k_d = sediment size coefficient.

This study focuses on developing a neural network model based on evolutionary training algorithm that pays attention not only to the prediction accuracy but also to the complexity of the network topology. The aim is to investigate the possibility of generating out of the optimised network a relatively simple but reasonably accurate formula for predicting the equilibrium scour depth around bridge piers.

2. Self-evolving network

Self-evolving networks are neural network variants whose training involves simultaneously optimising the synaptic weights and the network architecture. This type of network evolves automatically from the simplest topology consisting of a single hidden processing unit at the initial stage of training to a more complicated network depending on the complexity of the studied system. The self-evolution training algorithm involves the use of jumping particle optimization technique in updating the network topology while a combination of continuous version of particle swarm optimization (PSO) technique and method of least squares are used in fine-tuning the network constants. The algorithm helps reduce the user intervention while building the models by allowing the network structure to develop as the training progresses. The main advantage of the algorithm over the conventional neural network training is the production of more transparent and robust relationship between the inputs and the estimated parameter.

In this paper, scour data is used to develop an evolutionary network model, which is then used to generate a user-friendly formula for estimating the equilibrium depth of scour hole around cylindrical piers installed in a non-cohesive river bed.

2.1 Jumping particle swarm optimization (JPSO)

JPSO is a technique for handling combinatorial optimisation problems proposed by Matinez-Garcia Moreno-Perez [14]. The method is in some ways similar to the discrete particle swarm optimisation (DPSO) [15] with respect to the gravitation of particles towards more favourable positions. The key distinction between the two methods is that in the case of JPSO, the concept of “flying” is abandoned in favour of “jumping”, considering the discrete nature of the search space. The concept of jumping particle is based on the idea that in a discrete space a particle can only change its co-ordinate by jumping to a feasible position from the current one. Jumping to another position could be influenced by particles best experience, global most favourable

position, as well as their desire to explore unknown regions in the search space. The set of discrete parameters representing the co-ordinates of jumping particles are modified stochastically based on the influence of the above-mentioned factors. Eventually, regardless of the influencing factors, the particle jumps to the new co-ordinate only if it results in gaining some advantage over the current one. JPSO technique is implemented based on the following equation:

$$C_{t+1} = \lambda_1 \otimes C_t \oplus \lambda_2 \otimes \mathbf{b} \oplus \lambda_3 \otimes \mathbf{gb} \quad (1)$$

where C_t and C_{t+1} are the vectors of present and next particle positions in the combinatorial search space respectively. \otimes denotes the multiplication of a vector by a scalar, while \oplus refers to vector addition. The coefficients λ_1 , λ_2 and λ_3 are probabilities of random jumping, jumping towards the best particle co-ordinate and jumping towards best swarm position respectively. \mathbf{b} and \mathbf{gb} represent the particles' best and global most favourable co-ordinates respectively. The position updating implemented based on following scheme:

$$C_{i,t+1} = \begin{cases} C_{i,t} * \rho & P_{x_{i,t} \rightarrow \rho} = \lambda_1 \\ C_{i,t} * \mathbf{b}_i & P_{x_{i,t} \rightarrow \mathbf{b}_i} = \lambda_2 \\ C_{i,t} * \mathbf{gb}_i & P_{x_{i,t} \rightarrow \mathbf{gb}_i} = \lambda_3 \end{cases} \quad (2)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

ρ is a binary random number and P_x is the probability that the particle jumps towards a particular direction. The $*$ operation is executed by adjusting the topology of a given particle stochastically with some of the features of its attractor in a situation where it is attracted by either \mathbf{b} or \mathbf{gb} . If the particle, by chance, chooses to do an unguided jump, its topology will be modified by a randomly generated set of discrete parameters.

JPSO was successfully applied in some combinatorial optimisation problems including p-median problem [14], minimum labelling Steiner tree problem [16] and gear train benchmark problem [17]. In both [14] and [17], JPSO and the discrete version of PSO (DPSO) were compared, with the former returning better result in both cases.

2.2 Self-evolution training algorithm

Development of neural networks is a challenging task entailing not only optimising the weights of the synapses but also the architecture. The most challenging aspect of topology design is the control of the size of the network. Too many links and nodes will lead to redundancy and poor generalization. On the other hand, using insufficient neurons will lead to the inability of the network to extract the necessary information from the training data. The training algorithm used in this study attempts to tackle this challenge by optimizing the topology and unit weights at the same time. It involves carrying out the topology optimisation based on the JPSO technique and synaptic weight fine-tuning based on a combination of particle swarm optimization and the least squares methods. This algorithm handles neural network training as a mixed integer optimization problem, where a combination of discrete and continuous parameters is selected for maximum performance. To enhance computational efficiency, the algorithm is developed based on a bottom-to-top strategy, where the networks begin with single neuron at the hidden layer, then gradually develop into more complex topologies as they interact with the knowledge base during the training session. The first stage of the proposed self-evolution procedure involves generating a certain number of networks, each having a randomly generated synaptic links and synaptic weights. The binary connection parameters C_{ij} assume a value of 1 if the link between two nodes i and j exists and 0 otherwise (refer to Figure 1). In the course of training, the binary connection parameters are adjusted based on the jumping particle swarm optimisation algorithm, while the synaptic weights p_{ij} of connections between the input nodes and the non-linear hidden nodes are updated using the continuous PSO algorithm. With the output node being represented by a linear

model, the weights of connections to the output node, represented by $w_{i,j}$, are optimised using the method of least squares.

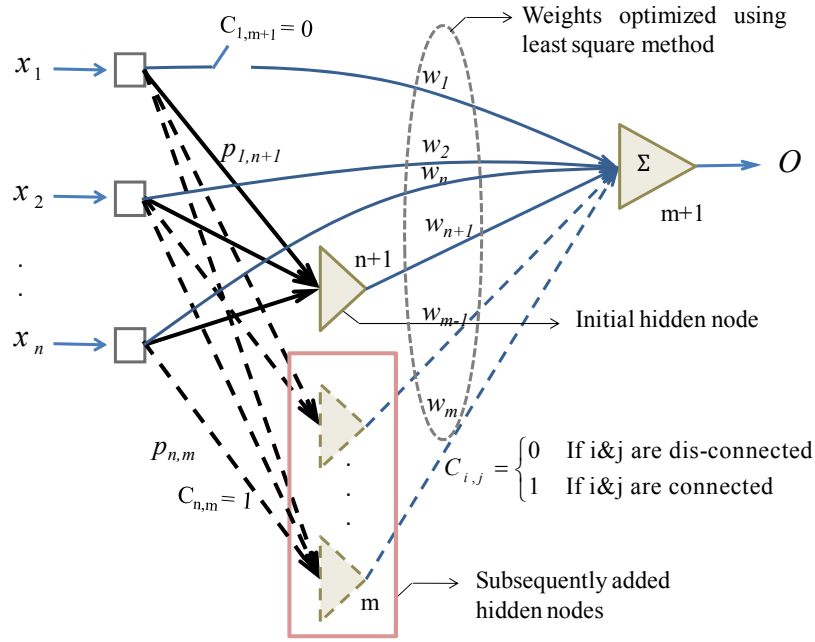


Fig. 1 - Topology of a Self-evolving Network

The size of the networks is gradually increased by adding more internal nodes, one node at a time. At the point of such nodal increment, only the information regarding the current particles best experience is retained by reducing the values of additional synaptic weights to zero and disconnecting the additional links. For the rest of network population, the synaptic weights and topologies are randomly reset. The rationale of retaining best results so far is to preserve the acquired knowledge as the optimisation process goes on while resetting the particle positions will help to avoid a premature convergence as the dimension of particles increase. The training algorithm is summarised as follows:

1. Randomly generate a population of N neural networks with single nodes in the hidden layer. Each network shall consist of a randomly generated set of connection parameters C and weights of synapses p .
2. Determine the synaptic weights w using the least squares method and evaluate the fitness of each network and update the best local and global fitness values and co-ordinates.
3. Use PSO/JPSO procedures to adjust the network topologies and weights for a certain number of iterations in accordance with the following steps:
 - Use the PSO algorithm to adjust the synaptic weight vector p of each particle (network).
 - Use the JPSO algorithm to adjust the connection parameters, C , of each particle (network).
 - Update the best local co-ordinate and the best global co-ordinate.
4. If sufficient convergence is achieved then go to 8. Else move on to 5.
5. Randomly reset the binary connection parameters and of duplicate particles (networks). Also, reset in the same manner the binary link switching parameters of certain fraction of the network population with poor fitness values. Go to 3.
6. If iterations number is not up to the maximum number go back to step3. Else move on.
7. Generate N networks (particles) with a number of internal nodes that is greater than the current number by one node. Replace all current networks with newly generated ones while retaining the global best co-ordinates (topology and synaptic weights). Go back to step 3.
8. Stop the algorithm and return result.

The algorithm is also summarised by the flowchart in Figure 2. To assist the population (swarm) of partially connected neuro-nets in the search for optimised network, a parallel swarm of fully connected neuro-nets but with the same number of hidden nodes is trained at the same time, with the former learning from the later whenever the best performing network lies in the latter.

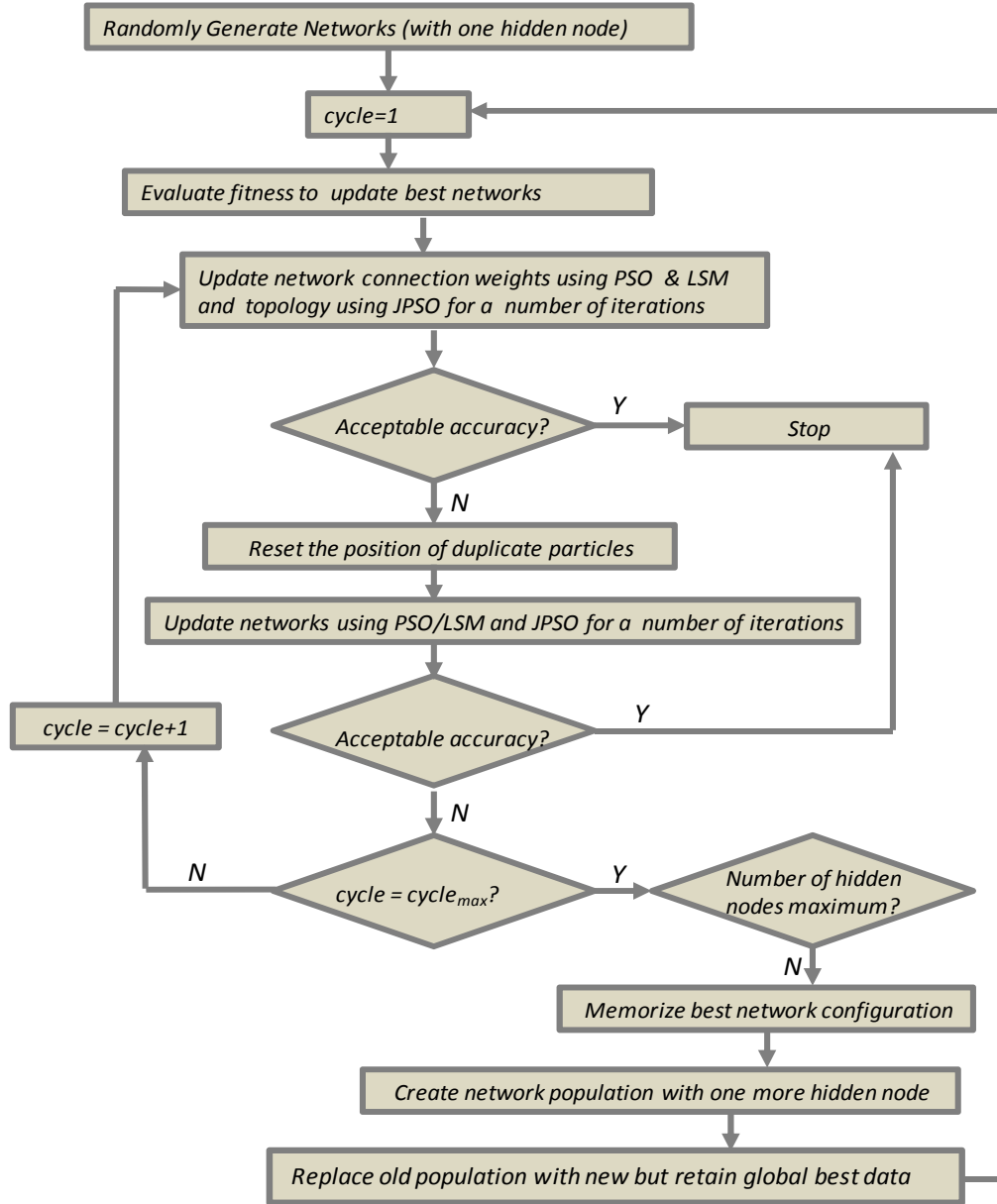


Fig. 2 - Network training flow diagram

2.3 Transfer function

The ability of networks to carry out a successful data mining depends significantly on the selection of transfer function type. For a given system, some transfer functions produce more accurate results with smaller network sizes than others. In this work, the product-unit function is used as a model for the hidden nodes in the network. Product-units process the products of weighted inputs they receive in contrast to summation units like sigmoid neurons. The choice of this function is informed by its greater information capacity, which tends to reduce the required number of hidden nodes [18]. The function is represented by the following expression:

$$f(x) = \prod_{i=1}^{i=n} x_i^{p_{i,j}} \quad (3)$$

where n is the number of input variables. x_i and p_{ij} are the input signals and the synaptic weights respectively. i is the input variable identifier, while j is the hidden node number. For instance $j=2$ refers to hidden node number 2. Thus p_{ij} is the exponent of x_i when the latter serves as input to hidden node j . In this work, the p values are limited to a range between -3 to +3 in order to prevent over-learning. Also for the sake of obtaining a relatively simple expression for the subsequent empirical model, the values of exponent weights, p , are rounded up to the nearest 0.25. It is important to remember, as indicated in Figure 1, that there are also direct links from the input nodes to the output representing the linear parts of the model, the strengths of which are optimized using the method of least squares.

3 Proposed model for scour depth prediction

3.1 Input parameters

The ability of neural networks to make reasonable predictions depends on how well the selected input parameters represent the factors controlling the behaviour of the studied system. The problem concerned in this study is the about the estimation of the scour depth around circular pier embedded into a uniform and cohesion-less river bed (see Figure 3). The factors controlling the depth of scour hole include the pattern of fluid flow, the properties of bed sediment, and the pier geometry [19]. The equilibrium scour depth can be expressed in terms of various controlling variables as:

$$d_{se} = f(\rho, \mu, U, Y, g, d_{50}, U_c, D) \quad (4)$$

in which ρ and μ are, respectively, the density and dynamic viscosity of the fluid. U and Y are the average velocity of approach and the depth of flow respectively. g , d_{50} , U_c and D are the acceleration due to gravity, the median diameter of bed sediment, the critical velocity respectively and the diameter of the pier. The eight independent parameters in Eq. (4) excluding density and viscosity can be lumped into a set of five dimension-less variables enclosed in the bracket in Eq. (5) as suggested in [19].

$$\frac{d_{se}}{D} = F\left(\frac{U}{U_c}, \frac{Y}{D}, \frac{U}{\sqrt{gY}}, \frac{d_{50}}{D}\right) \quad (5)$$

In this study, the network is trained to predict the normalised scour depth based on Equation (5). The $\tanh()$ transformations of the inputs are also added as inputs to the network to facilitate a reduction in the number of network constants. The use of non-dimensional parameters in this work is informed by the need to do away with scale effects on the predicted quantity. It is noteworthy that Equation (5) satisfies the Buckingham pi theorem [], as it comprises of solely non-dimensional parameters. For instance, U and U_c are of the same dimension (LT^{-1}), therefore U/U_c is dimension-less.

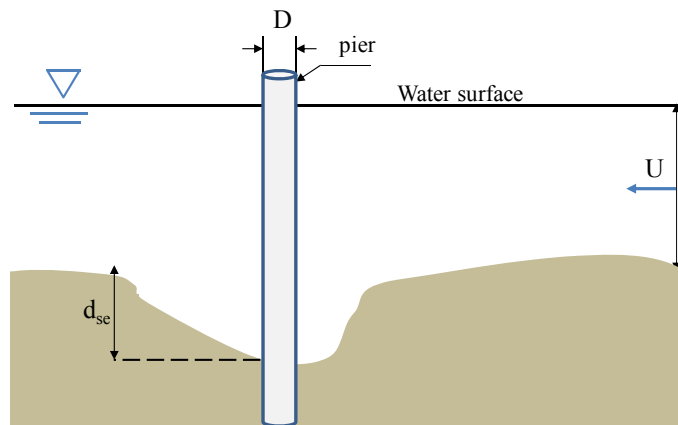


Fig. 3 - Scour around a circular pier

3.2 Database description

The database compiled for this work is the same scour data used by Bateni et al. [8], which consists of scour depth measurements for a given flow velocity, pier diameter, median particle size and depth of flow. The database comprises of 269 data points covering a wide range of pier diameters, regimes of flow and median particle sizes. The summary of characteristics of the database is given in Table 2.

Table 2

Database statistical properties						
	Input parameter					
	d_{se} (m)	D (m)	d_{50} (mm)	U (m/s)	U_c (m/s)	Y (m)
μ	124.08	197.87	1.525	0.3897	0.431,8	126.48
σ	77.99	219.40	1.353	0.2409	0.1737	97.22
x_{max}	440.00	1000.00	7.80	1.50	1.25	600.00
x_{min}	4.00	10.00	0.20	0.17	0.19	20.00

3.3. Network training and testing

To develop the proposed model, the database was divided into training and testing sets. Approximately 42% of the database (113 data points) was used for the sake of training, while the remaining 58% of the data were reserved for testing. The reason for allocating nearly half of the data for testing is to ensure a more robust assessment of the model's accuracy given the limited number of data points. In order for the network to capture the widest variations in input and output patterns in the database, the maximum and minimum values in the database are included in the training data. This is to avoid having extreme data points in the testing set, which could undermine the assessment of true generalization capability of the model. For the sake of comparison, conventional back-propagation network with a transfer function based on a combination of sigmoid and linear activation functions was also trained using the same data. The self-evolving network was trained in accordance with the optimisation algorithm described in subsection 2.1. To guard against over-learning and unnecessary network complexity, the learning criterion is based on the cost function proposed in [21], which penalises both topology complexity and inaccuracy in prediction. The key advantage of this method over alternative approaches such as weight decay method is that the former focuses on minimising the number of links in the network rather than decreasing the magnitude of the synaptic weights, which often doesn't help in reducing the network complexity. Other metrics used in assessing the prediction accuracy of the proposed model include the non-dimensional root mean square error (N-RMSE) and the coefficient of determination (R2).

$N\text{-RMSE}$ is defined by the following expression:

$$N - RMSE = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{i=n} (O_i - y_i)^2}}{O_{rms}} \quad (6)$$

The coefficient of determination is defined as:

$$R^2 = \frac{(n \sum y_i O_i - (\sum y_i)(\sum O_i))^2}{(n(\sum y_i^2) - (\sum y_i)^2) \cdot (n(\sum O_i^2) - (\sum O_i)^2)} \quad (7)$$

where O_i , y_i and O_{rms} are, respectively, the experimental value, the predicted value, and the root mean square value of the observed data.

4 Results and discussions

4.1 Training and testing results

The training and testing results of the proposed model are displayed in Figures 4(a) and 4(b) respectively. As the figures showed, the predictions of the network are in a good agreement with the scour data, given the reasonably high value of R^2 in both cases of training data ($R^2 = 0.856$) and the testing data ($R^2 = 0.810$). This is indicative of success of the learning process undergone by the networks not only because the network parameters are fine-tuned enough to yield a good approximation of the experimental observations, but more importantly because of the ability of the networks to produce estimates of scour depth comparison with the data that was not part of the training set. The results presented so far are based on 215 data points. The remaining 53 points are considered as outliers, considering the huge difference (at least 100%) between the measurements and the predictions of various methods considered in this work. They are therefore excluded from training and testing data sets.

The prediction accuracy of the proposed model is further examined by plotting the values of residuals (predicted value less the actual value) against the test data (see Figure 5). It can be seen from the figure that the data points are highly scattered around the horizontal axis and no clear trend can be identified. This suggests that the prediction errors are more likely to be as a result of noise rather than model shortcomings.

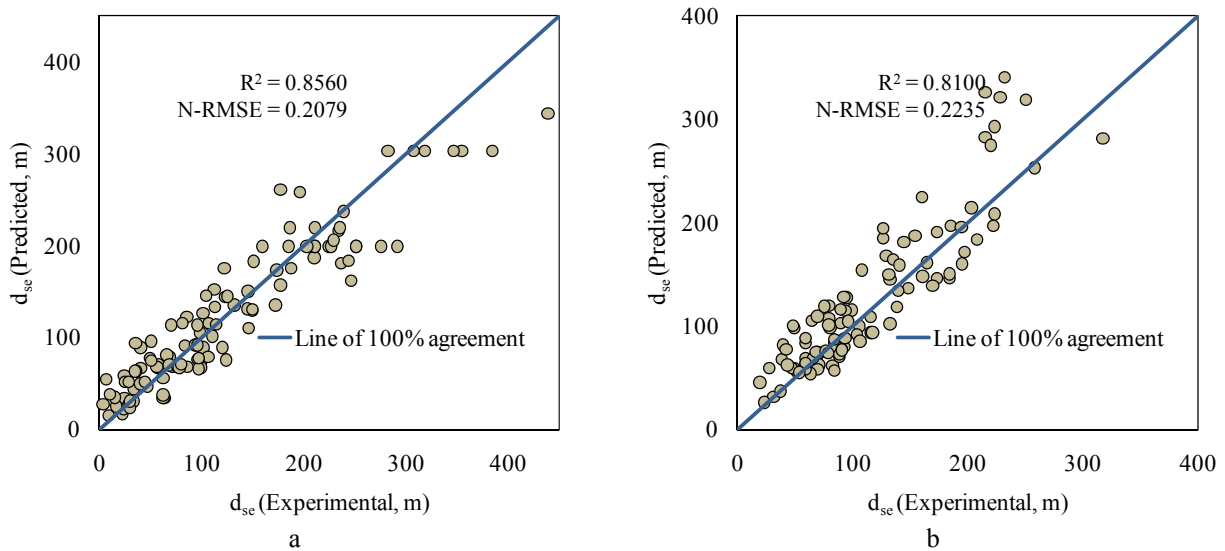


Fig. 4 - Comparison of proposed model predictions with experimental observations
(a) Training set and (b) testing data set

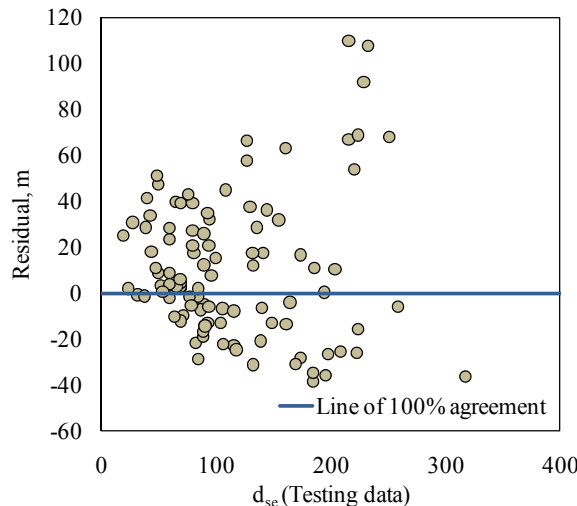


Fig. 5 - Residuals versus testing data

4.2 Parametric study

The parametric study is meant to assess the conformity of model predictions with the underlying behaviour of the system studied. The task is carried out by adopting a commonly used technique [22,23], which entails varying the subject variable between the minimum and the maximum values in the database while keeping the rest of the input variables to values around the average. In this way, the desired variables are investigated, one at a time. The results of the study carried out on the proposed model by varying the ratio of median sediment size (d_{50}) to pier diameter and the ratio of flow depth (Y) to pier diameter (D) are plotted in Figures 6-7. The figures showed that the proposed model behaved in accordance with the known relationship between the scour depth and its controlling factors [19a]. The consistent decrease in the predicted scour depth (per unit width of pier) with increment in median particle size in Figure 6 indicates a good agreement between the estimates of the proposed model and the expected response of sand bed, where the eroded quantity is inversely related with the particle size. The positive variation of non-dimensional scour depth with normalized flow depth (Y/D) observed Figure 7 is also a rational response by the model as it is in tandem with the current knowledge regarding the mechanism of scour hole evolution.

Sensitivity analysis has also been carried out to study the strength of the effects of various input parameters on the proposed model. To test a given parameter, it is disconnected from the network and the prediction quality of the model is assessed. The results of the analysis are summarized in Table 3, which indicated that all input parameters considered in this model significantly affect the prediction quality of the developed model.

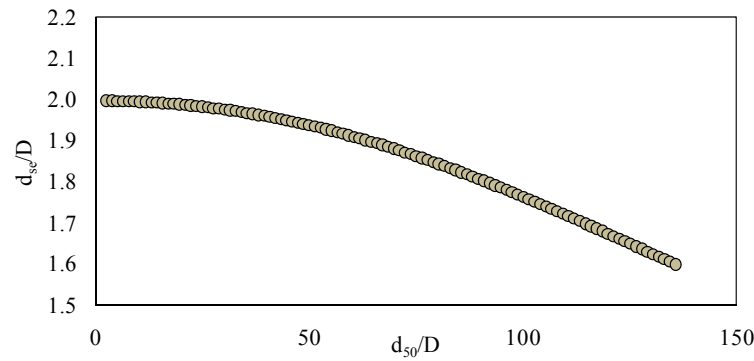


Fig. 6 - Variation of non-dimensional equilibrium scour depth with d_{50}/D ($\times 10^{-3}$)

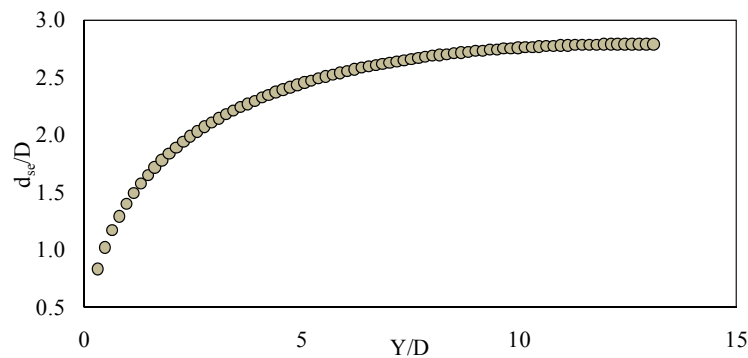


Fig. 7 - Variation of non-dimensional equilibrium scour depth with Y/D

Table 3

Results of sensitivity study	
Excluded parameter	N -RMSE (testing)
None	0.2235
d_{50}/D	38.200
U/U_c	2.116
Y/D	0.4623
F_r	0.4784

4.2 Comparisons with existing models

Based Previous studies [8,9], the existing empirical models, despite varying in complexity as Table 1 showed, generally suffer from large prediction errors when compared with experimental data. In this section, the prediction quality of the proposed model in estimating the equilibrium scour depth is further assessed by comparing its results are compared with that of the back-propagation (BP) neural network as well as some of the empirical models available in the literature. From the results presented in Table 4, it is clear that the BP network produces less accurate estimates with respect to testing data than the proposed model despite using a far greater number of constants (60). In previous works based on Artificial Neural Network (ANN), a greater accuracy was reported but at a cost of a large number of parameters. For instance, the models developed by Bateni et al.[8] and Ismail et al.[9] have 128 constants and 70 constants respectively. With 17 constants, the proposed model is clearly much less complicated than both the models developed in [8] and [9]. The existing empirical models summarised in Table 1 produced much less accurate results than both BP network and the proposed models with the Hancu's model [3] being the worst predictor ($R^2 = 0.3247$). The inferior prediction quality of conventional formulae seen here corroborates the findings of the previous researches on the subject [8,9,24,25].

Table 4

Prediction results of various methods

Model type	No. of constants	N -RMSE (testing)	R^2 (testing)
Proposed formula	17	0.2235	0.8100
BPN model	60	0.3057	0.7084
Laursen and Toch [1]	-	0.3615	0.6408
Hancu [3]	-	0.7259	0.3247
US DoT [5]	-	0.4901	0.4947
Melville and Chiew [6]	-	0.3382	0.6804

In summary, the proposed model turns out to be the best among the models tested, having returned the least prediction error with a relatively small number of constants. The following formula represents the selected model:

$$\frac{d_w}{D} = \left(k_1 - k_2 \left(\frac{Y}{D} \right)^{0.25} \right) \left(\frac{Y}{D} \right)^{0.5} F_r^{0.25} - k_3 \quad (7)$$

The parameters k_1 , k_2 and k_3 are defined as follows:

$$\begin{aligned} k_1 &= 577.1597 \left(\frac{D}{d_{50}} \right) \tanh \left(\frac{d_{50}}{100D} \right) \tanh \left(\frac{U}{2U_c} \right) \left(\frac{U}{U_c} \right)^{-0.25} \\ k_2 &= 2976.040 \left(\frac{D}{d_{50}} \right)^{1.75} \left[\tanh \left(\frac{d_{50}}{100D} \right) \right]^{1.75} \left[\tanh \left(\frac{U}{2U_c} \right) \right]^{-0.25} \left(\frac{U}{U_c} \right)^{0.25} \\ k_3 &= 14.4648 \left(\frac{D}{d_{50}} \right)^{0.75} \left[\tanh \left(\frac{d_{50}}{100D} \right) \right]^{0.75} \left[\tanh \left(\frac{U}{2U_c} \right) \right]^{0.75} \left(\frac{U}{U_c} \right)^{1.25} - 0.0064 \end{aligned} \quad (8)$$

where d_{50} is in mm; D and Y are in m; U and U_c are in m/s.

5 Conclusions

Self-evolving network training algorithm was used to build an empirical model for estimating the equilibrium depth of scour around circular piers. The choice of training algorithm is based on the desire to produce a model capable of making reasonably accurate scour predictions with a relatively small number of constants. The proposed model is based on experimental data comprising of a broad range of pier diameters, median particle sizes and flow depths and velocities.

Based on the results obtained, the proposed formula correlates better with validating experimental data than BPN network and existing formulae for scour depth predictions. The formula also benefits from much fewer constants in comparison to the BP networks.

References

- [1] Laursen, E.M. and Toch, A. (1956). *Scour around bridge piers and abutments*, Iowa Road Res. Board, 4:60.
- [2] Shen H.W. (1971). *River mechanics*. New York, USA: John Wiley and Sons, 1971, 2: 23.
- [3] Hancu S (1971). Sur le calcul des affouillements locaux dans la zone des piles des ponts. Proc. 14th IAHR Congress. Paris, France, 3, 299-313.
- [4] Breusers H. N. C., Nicollet G. and Shen H. W. (1977). Local scour around cylindrical piers. *J. Hydr. Res.*, 15(3), 211-252. DOI: 10.1080/00221687709499645
- [5] U. S. DoT (1992). *Evaluating scour at bridges*, hydraulic Engineering circular 18, Federal Highway administration, 2, 1-4.
- [6] Melville B. W. and Chiew Y. M. (1999). Time scale for local scour depth at bridge piers. *J. Hydr. Eng.*, 125(1), 59-65. DOI: 10.1061/(ASCE)0733-9429(1999)125:1(59)
- [7] Lee, T. L., Jeng, D. S., Zhang, G. H. and Hong, J. H. (2007). Neural network modeling for estimation of scour depth around bridge piers, *Journal of Hydrodynamics*, 19(3), 378-386. DOI: 10.1016/S1001-6058(07)60073-0
- [8] Bateni, S.M., Borghei, S.M. and Jeng, D.S. (2007). Neural network and neuro-fuzzy assessments for scour depth around bridge piers, *Engineering Applications of Artificial Intelligence*, 20 (3), 401-414. DOI: 10.1016/j.engappai.2006.06.012
- [9] Ismail A., Jeng D.-S., Zhang, L.L. and Zhang, J.-S. (2013). Predictions of bridge scour: Application of a feed-forward neural network with an adaptive activation function, *Engineering Applications of Artificial Intelligence*, 26, 1540–1549. DOI: 10.1016/j.engappai.2012.12.011
- [10] Akib, S., Mohammadhassani, M. and Jahangirzadeh, A. (2014). Application of ANFIS and LR in prediction of scour depth in bridges, *Computers & Fluids*, 91, 77-86, DOI: 10.1016/j.compfluid.2013.12.002
- [11] Ebtehaj, I., Hossein Bonakdari, H., Moradi, F., Gharabaghi, B. And Khozani, Z. S. (2018). An integrated framework of Extreme Learning Machines for predicting scour at pile groups in clear water condition, *Coastal Engineering*, 135, 1-15. DOI: 10.1016/j.coastaleng.2017.12.0123
- [12] Chou, J.-S. And Pham, A.-D. (2014). Hybrid computational model for predicting bridge scour depth near piers and abutments, *Automation in Construction*, 48, 88-96. DOI: 10.1016/j.autcon.2014.08.006
- [13] Pang, A. L. J., Skote, M., Lim, S.Y., Gullman-Strand, J. and Morgan, N. (2016). A numerical approach for determining equilibrium scour depth around a mono-pile due to steady currents, *Applied Ocean Research*, 57, 114-124. DOI: 10.1016/j.apor.2016.02.010.
- [14] Matinez-Garcia F. J. and Moreno-Perez, J. A. (2008). *Jumping frogs optimization: a new swarm method for discrete optimization*, Technical Report DEIOC 3/2008, Department of Statistics, O.R. and Computing, University of La Laguna, Tenerife, Spain.
- [15] Kennedy, J. and Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm, Proceedings of IEEE Conference on Systems, Man, and Cybernetics, Piscataway, New Jersey, USA, p. 4104-4109.
- [16] Consoli, S., Moreno-Perez, J. A., Darby-Dowman, K. and Mladenovic, N. (2010). Discrete particle swarm optimisation for the minimum labelling steiner tree problem, *Natural Computing*, 9(1), pp. 29–46.
- [17] Seren, C. (2011). A hybrid jumping particle swarm optimisation method for high dimensional unconstrained discrete problems, IEEE Congress on Evolutionary Computation (CEC), p. 1649-1656, 2011.
- [18] Durbin R. and Rumelhart, R. (1989). Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural computation*, 1, 133–142. DOI: 10.1162/neco.1989.1.1.133
- [19] Melville, B. W., & Chiew, Y. M. (1999). Time scale for local scour at bridge piers. *Journal of Hydraulic Engineering*, 125(1), 59-65.
- [20] Buckingham, E. (1914). On physically similar systems; illustrations of the use of dimensional equations". *Physical Review*. 4 (4): 345–376.
- [21] Jin, Y., Okabe, T. and Sendhoff, B. (2004). Neural network regularization and ensembling using multi-objective evolutionary algorithms, Congress on Evolutionary Computation (CEC'04), IEEE Press, 1–8.
- [22] Goh, A. T. C. (1995). Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering*, (9):143–151.
- [23] Rahman, M. S., Wang, J., Deng, W. and Carter, J. P. (2001). A neural network model of the uplift capacity of suction caissons. *Computers and Geotechnics*, 28:269–287.
- [24] Firat, M. and Gungor, M. (2009). Generalized regression neural networks and feed forward neural networks for prediction of scour depth around bridge piers. *Adv. Eng. Software*, 40, 731–737. DOI: 10.1016/j.advengsoft.2008.12.001
- [25] Abidin, K. (2010). Artificial neural network study of observed pattern of scour depth around bridge piers. *Comput. Geotech.* 37, 413–418. DOI: 10.1016/j.compgeo.2009.10.003