TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 6(55) (2022) 16-22
TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

Artificial neural network for regression analysis developed in excel VBA: a case study in pile bearing capacity prediction

Mô hình mạng nơ-ron thần kinh nhân tạo dùng cho phân tích hồi quy được phát triển trong Excel VBA: Ứng dụng cho dự báo sức chịu tải của cọc

Hoang Nhat Duc[a,b*]
Hoàng Nhật Đức[a,b*]

[a]*Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam*
[a]*Viện Nghiên cứu và Phát triển Công nghệ Cao, Đại học Duy Tân, Đà Nẵng, Việt Nam*
[b]*Faculty of Civil Engineering, Duy Tan University, Da Nang, 550000, Vietnam*
[b]*Khoa Xây dựng, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam*

(Ngày nhận bài: 22/6/2022, ngày phản biện xong: 04/8/2022, ngày chấp nhận đăng: 15/10/2022)

Abstract

This paper aims at developing an artificial neural network (ANN) for regression analysis. The ANN model is developed in Excel Visual Basic for Applications (VBA) to facilitate its practical implementations. The capability of the developed ANN program has been tested with the task of pile bearing capacity prediction.

Keywords: Artificial neural network; Data analysis; Regression analysis; Excel VBA; Pile bearing capacity.

Tóm tắt

Bài báo xây dựng một công cụ dùng cho phân tích hồi quy dựa trên mô hình mạng nơ-ron thần kinh nhân tạo (ANN). Mô hình ANN được phát triển trên nền tảng Excel VBA để nâng cao tính ứng dụng thực tiễn. Chương trình phân tích hồi quy dựa trên ANN đã được sử dụng để dự báo sức chịu tải của cọc.

Từ khóa: Mạng nơ-ron thần kinh nhân tạo; phân tích dữ liệu; Phân tích hồi quy; Excel VBA; Sức chịu tải của cọc.

1. Introduction

Regression analysis is a crucial task in the construction industry. Regression models are able to analyze past data records and yield prediction results that immensely help the decision-making processes during various phases of a construction project. Various regression analysis methods have been successfully developed to cope with a wide range of problems such as pile bearing capacity estimation [1-6], concrete strength estimation [7-11], the soil compression coefficient [12, 13], soil shear strength modeling [14-16], etc.

Particularly, the ANN-based regression model plays an important role in modeling various complex phenomena in civil

**Corresponding Author:* Hoang Nhat Duc, Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam; Faculty of Civil Engineering, Duy Tan University, Da Nang, 550000, Vietnam.
Email: hoangnhatduc@duytan.edu.vn

engineering [17, 18]. Nevertheless, few studies have dedicated to the development of ANN models within the Excel VBA environment. Microsoft Excel is a popular tool for performing various modeling tasks in civil engineering [19-21]. In this regard, the ability of building regression analysis models in Excel can be helpful for practicing engineers who have to cope with various data analysis tasks. This paper contributes a tool implementing the ANN regression model that can be directly built and used in the Excel VBA environment.

2. Artificial Neural Network (ANN) developed in Excel VBA

An ANN regression model relies on a set of neurons in the hidden layer to compute the predicted result of a dependent variable [22]. An ANN model consisting of two neurons is illustrated in **Fig. 2.1**. To cope with nonlinear regression tasks, an activation function (e.g. the sigmoid function) can be used [23]. The process of model training and prediction using an ANN model can be described as follows (i) Data collection, (ii) Data normalization, (iii) Model selection, (iv) Model training, and (v) Model prediction and evaluation.

In the first step, a dataset including predictor variables and predicted variable is collected. The ranges of the variables in this dataset are standardized in the second step. Usually, the *Z*-score equation is used. The suitable parameters of an ANN model including the learning rate and the number of neurons in the hidden layer are determined in the model selection step. The model is trained in the next step. The final step involves the employment of the trained model to predict novel data samples. Various indices such as root mean square error (RMSE), mean absolute percentage error (MAPE), and coefficient of determination (R^2) can be used to quantify the model performance [24]. An ANN class is coded in Excel VBA and its object can be created for constructing a regression model (refer to **Fig. 2.2**). The ANN class utilizes supporting functions stored in a class named myMatrix (refer to **Fig. 2.3**).

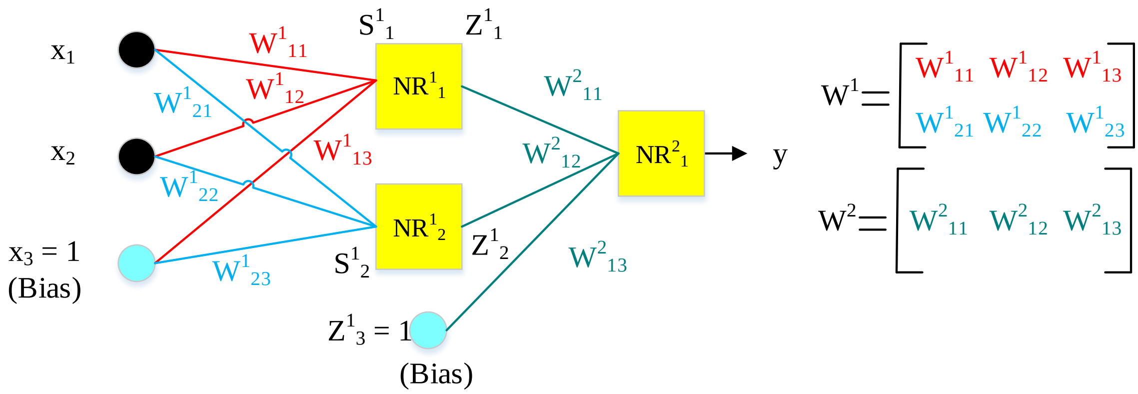


Fig. 2.1 Demonstration of an ANN consisting of two neurons in the hidden layer

```
Dim model As ANNR
Set model = New ANNR
model.NR = 10
model.alpha = 0.002
model.MaxEpoch = 500
model.Set_X (Xtr)
model.Set_T (Ttr)
model.Train
Dim W1_Trained As Variant, W2_Trained As Variant
W1_Trained = model.Get_W1
W2_Trained = model.Get_W2
```

(a)

```
' Class: Artificial Neural Network for Regression
Private X As Variant ' Training features
Private T As Variant ' Training target
Private W1 As Variant ' Weight matrix of the hidden layer
Private W2 As Variant ' Weight matrix of the output layer
Private N As Integer ' Number of training samples
Private D As Integer ' Number of features
Public NR As Integer ' Number of neuron
Public alpha As Double ' Learning Rate
Public MaxEpoch As Integer
Public RMSE, MAPE, R2 As Double
Public Function Train()
    Dim mM As myMatrix
    Set mM = New myMatrix
    D = UBound(X, 2)
    W1 = mM.RandomMatrix(NR, D + 1, -3, 3)
    W2 = mM.RandomMatrix(1, NR + 1, -3, 3)
```

(b)

Fig. 2.2 The ANN model developed in Excel VBA: (a) Create an ANN object and (b) The function used for model training

```
Public Function RandomMatrix(ByVal N As Integer, ByVal D As Integer, ByVal LB As Double, _
        ByVal UB As Double) As Variant
    Dim W As Variant
    ReDim W(N, D)
    Dim i, k As Integer
    For i = 1 To N
        For k = 1 To D
            W(i, k) = LB + (UB - LB) * Rnd()
        Next k
    Next i
    RandomMatrix = W
```

(a)

```
Public Function Multiplication(ByVal A As Variant, B As Variant) As Variant
    ' A = MxN. B = NxP. C = MxP
    Dim M, N, P As Integer
    M = UBound(A, 1)
    N = UBound(A, 2)
    P = UBound(B, 2)
    Dim C As Variant
    ReDim C(M, P)
    Dim i, j, k As Integer

    For i = 1 To M
        For j = 1 To P
            C(i, j) = 0
            For k = 1 To N
                C(i, j) = C(i, j) + A(i, k) * B(k, j)
            Next k
        Next j
    Next i
    Multiplication = C
End Function
```

(b)

Fig. 2.3 Examples of functions provided in the myMatrix class: (a) The function that generates a matrix of random numbers and (b) The function that performs matrix multiplication

```
' Backward pass --------------------
Dim Delta As Variant
ReDim Delta(N, 1)
Dim k As Integer
For k = 1 To N
    Delta(k, 1) = T(k, 1) - Y(k, 1)
Next k
Dim Z1_B_k As Variant
Dim dE_dW2 As Variant
Dim dE_dW1 As Variant
For k = 1 To N
    Z1_B_k = mM.ExtractMatrixColumn(Z1_B, k)
    Z1_B_k = mM.Transpose(Z1_B_k)
    dE_dW2 = mM.MultiplyWithScalar(Z1_B_k, -1 * Delta(k, 1))
    Dim Q As Variant
    ReDim Q(NR, 1)
    Dim v As Integer
    For v = 1 To NR
        Q(v, 1) = -Delta(k, 1) * W2(1, v) * Z1_B(v, k) * (1 - Z1_B(v, k))
    Next v
    Dim XM As Variant
    ReDim XM(NR, D + 1)
    Dim z As Integer
    For v = 1 To NR
        For z = 1 To D + 1
            XM(v, z) = X_B(k, z)
        Next z
    Next v
    dE_dW1 = mM.Multiplication(Q, XM)
    ' Update W1 and W2
    For v = 1 To NR
        For z = 1 To D + 1
            W1(v, z) = W1(v, z) - alpha * dE_dW1(v, z)
        Next z
    Next v
    For v = 1 To NR + 1
        W2(1, v) = W2(1, v) - alpha * dE_dW2(1, v)
    Next v
Next k
```

Fig. 2.4 VBA code used to adapt the ANN’s weights

Given a dataset $\{X_i, T_i\}_{i=1,2,...,N}$, $X_i = [X_1, X_2, ..., X_{D+1}]$ denoting the i^{th} predictor variable, and T_i representing the i^{th} dependent variable, an ANN model can be trained by this dataset to approximate the nonlinear mapping from X_i to T_i. It is noted that $X_{D+1} = 1$ corresponds to the bias in the input layer. Given the matrices W_1 and W_2, this mapping function can be described as follows:

$$S_u^1 = \sum_{k=1}^{D+1} W_{1k}^1 \times x_k \quad (1)$$

where u = 1,2,…, NR denotes the index of a neuron in the hidden layer. NR is the number of neurons.

$$Z_u^1 = \sigma(S_u^1) = \frac{1}{1+\exp(-S_u^1)} \quad (2)$$

where $\sigma()$ is the sigmoid activation function.

$$y = \sum_{u=1}^{NR+1} Z_u^1 \times W_{1u}^2 \quad (3)$$

where $u = 1,2,...,NR+1$ and $Z_{NR+1}^1 = 1$ accounts for the bias in the hidden layer.

The weight matrices of an ANN model are adapted via the backpropagation and gradient descent algorithms [25]. The VBA code used to adapt the ANN's weights (W^1 and W^2) is demonstrated in **Fig. 2.4**. Given the loss function $E = \frac{1}{2}(t-y)^2$, the equations used to update the weight matrices:

$$W_{i,j}^L = W_{i,j}^L - \alpha \times \frac{\partial E}{\partial W_{i,j}^L}$$

where L denote the index of layers in the ANN model.

The partial derivative of E with respect to a weight of W_2 is given by:

$$\frac{\partial E}{\partial W_{1u}^2} = \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial W_{1u}^2} = -(t-y) \times Z_1^1 = -\Delta \times Z_u^1 \quad (4)$$

Alternatively, $\frac{\partial E}{\partial W^2} = -\Delta \times [Z_1^1, Z_2^1, ..., 1]$ (5)

Considering an ANN model with two neurons and two input variables, the equation used to compute the partial derivative of E with respect to a weight in the first row of W_1 is given by:

$$\begin{aligned}\frac{\partial E}{\partial W_{11}^1} &= \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial Z_1^1} \times \frac{\partial Z_1^1}{\partial S_1^1} \times \frac{\partial S_1^1}{\partial W_{11}^1} = -(t-y) \times W_{11}^2 \times [Z_1^1 \times (1-Z_1^1)] \times x_1 \\ &= -\Delta \times W_{11}^2 \times [Z_1^1 \times (1-Z_1^1)] \times x_1 \end{aligned} \quad (6)$$

where $\frac{\partial Z_1^1}{\partial S_1^1} = [Z_1^1 \times (1-Z_1^1)]$.

$$\begin{aligned}\frac{\partial E}{\partial W_{12}^1} &= \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial Z_1^1} \times \frac{\partial Z_1^1}{\partial S_1^1} \times \frac{\partial S_1^1}{\partial W_{12}^1} = -(t-y) \times W_{11}^2 \times [Z_1^1 \times (1-Z_1^1)] \times x_2 \\ &= -\Delta \times W_{11}^2 \times [Z_1^1 \times (1-Z_1^1)] \times x_2 \end{aligned} \quad (7)$$

$$\begin{aligned}\frac{\partial E}{\partial W_{13}^1} &= \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial Z_1^1} \times \frac{\partial Z_1^1}{\partial S_1^1} \times \frac{\partial S_1^1}{\partial W_{13}^1} = -(t-y) \times W_{11}^2 \times [Z_1^1 \times (1-Z_1^1)] \times x_3 \\ &= -\Delta \times W_{11}^2 \times [Z_1^1 \times (1-Z_1^1)] \times 1 \end{aligned} \quad (8)$$

Similarly, the partial derivative of E with respect to a weight in the second row of W_1 is given by:

$$\begin{aligned}\frac{\partial E}{\partial W_{23}^1} &= \frac{\partial E}{\partial y} \times \frac{\partial y}{\partial Z_2^1} \times \frac{\partial Z_2^1}{\partial S_2^1} \times \frac{\partial S_2^1}{\partial W_{23}^1} = -(t-y) \times W_{12}^2 \times [Z_2^1 \times (1-Z_2^1)] \times x_3 \\ &= -\Delta \times W_{12}^2 \times [Z_2^1 \times (1-Z_2^1)] \times 1 \end{aligned} \quad (9)$$

Alternatively, the weights of W_1 are updated as follows:

$$\frac{\partial E}{\partial W_1^1} = -\Delta \times W_{11}^2 \times \{Z_1^1 \times (1 - Z_1^1)\} \times [x_1, x_2, x_3] = -\Delta \times W_{11}^2 \times \{Z_1^1 \times (1 - Z_1^1)\} \times [x_1, x_2, 1] \quad (10)$$

$$\frac{\partial E}{\partial W_2^1} = -\Delta \times W_{12}^2 \times \{Z_2^1 \times (1 - Z_2^1)\} \times [x_1, x_2, x_3] = -\Delta \times W_{12}^2 \times \{Z_2^1 \times (1 - Z_2^1)\} \times [x_1, x_2, 1] \quad (11)$$

Accordingly, it is able to summarize the updating rule for W_1 as follows:

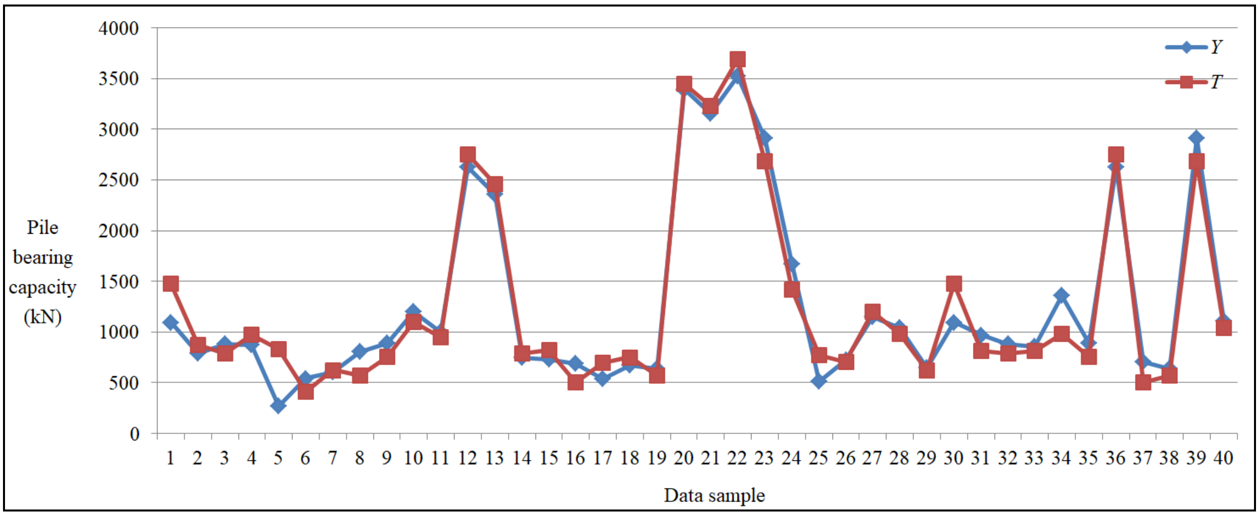
$$\frac{\partial E}{\partial W^1} = \begin{bmatrix} a \\ b \end{bmatrix} \times [x_1, x_2, 1] \quad (12)$$

where $a = -\Delta \times W_{11}^2 \times \{Z_1^1 \times (1 - Z_1^1)\}$ and $b = -\Delta \times W_{12}^2 \times \{Z_2^1 \times (1 - Z_2^1)\}$.

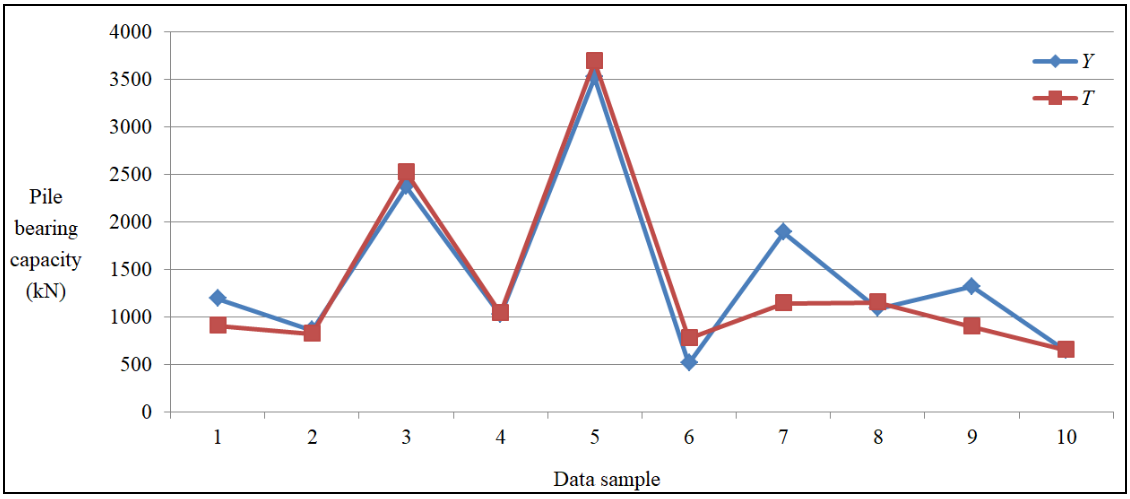
3. Model application

In this section, the ANN regression model constructed in Excel VBA is used to estimate the pile bearing capacity. The ANN model is trained and tested with a dataset collected in [26]. Herein, the hammer weight (kN), drop height (m), length (m), pile set (mm), and pile cross-sectional area (cm^2) are used as predictor variables. The pile bearing capacity measured in kN is the dependent variable. There are 50 records in the collected data. Herein, 40 samples have been used for training the ANN. The rest of the data is used for testing the trained model.

With the number of neurons = 10, the learning rate = 0.002, and the number of training epochs = 500, the ANN can successfully learn the mapping function between the set of predictor variables and the pile bearing capacity. The RMSE, MAPE, and R^2 of the model in the training phase are 186.42, 18.35%, and 0.96. These indices in the testing phase are 306.56, 17.06%, and 0.89. These facts mean that the ANN model is able to explain 96% and 89% of the variation of the pile bearing capacity in the training and testing datasets, respectively. The prediction results of the ANN model are graphically presented in **Fig. 3.1**.



(a)



(b)

Fig. 3.1 The prediction results of the ANN model (*Y* and *T* denote the predicted and actual pile bearing capacity, respectively): (a) Training phase and (b) Testing phase

4. Conclusion

Regression analysis is an important task in civil engineering. This study has put forward an ANN model developed in Excel VBA to assist the decision-making process involving approximation of nonlinear functions. The ANN model has been used to estimate the pile bearing capacity with good prediction performance (R^2 = 0.89 in the testing phase). Future extensions of the current study may consider the applications of the ANN model in other nonlinear regression tasks.

Supplementary material

The Excel VBA code of the program can be accessed at:

https://github.com/NDHoangDTU/ANNR_ExcelVBA

References

[1] N.-D. Hoang, X.-L. Tran, T.-C. Huynh (2022), *Prediction of Pile Bearing Capacity Using Opposition-Based Differential Flower Pollination-Optimized Least Squares Support Vector Regression (ODFP-LSSVR).* Advances in Civil Engineering 2022:7183700. doi:10.1155/2022/7183700

[2] H. Harandizadeh, D. Jahed Armaghani, M. Khari (2021), *A new development of ANFIS–GMDH optimized by PSO to predict pile bearing capacity based on experimental datasets.* Engineering with Computers 37 (1):685-700. doi:10.1007/s00366-019-00849-3

[3] T.A. Pham, V.Q. Tran, H.-L.T. Vu, H.-B. Ly (2020), *Design deep neural network architecture using a genetic algorithm for estimation of pile bearing capacity.* PLOS ONE 15 (12):e0243030. doi:10.1371/journal.pone.0243030

[4] H. Harandizadeh, V. Toufigh (2020), *Application of Developed New Artificial Intelligence Approaches in Civil Engineering for Ultimate Pile Bearing Capacity Prediction in Soil Based on Experimental Datasets.* Iranian Journal of Science and Technology, Transactions of Civil Engineering 44 (1):545-559. doi:10.1007/s40996-019-00332-5

[5] H. Harandizadeh (2020), *Developing a new hybrid soft computing technique in predicting ultimate pile bearing capacity using cone penetration test data.* Artificial Intelligence for Engineering Design, Analysis and Manufacturing 34 (1):114-126. doi:10.1017/S0890060420000025

[6] B. Ghorbani, E. Sadrossadat, J. Bolouri Bazaz, P. Rahimzadeh Oskooei (2018), *Numerical ANFIS-Based Formulation for Prediction of the Ultimate Axial Load Bearing Capacity of Piles Through CPT Data.* Geotechnical and Geological Engineering 36 (4):2057-2076. doi:10.1007/s10706-018-0445-7

[7] H. Nguyen, T. Vu, T.P. Vo, H.-T. Thai (2021), *Efficient machine learning models for prediction of concrete strengths.* Construction and Building Materials 266:120950. doi: https://doi.org/10.1016/j.conbuildmat.2020.120950

[8] J.-S. Chou, C.-F. Tsai, A.-D. Pham, Y.-H. Lu (2014), *Machine learning in concrete strength simulations: Multi-nation data analytics.* Construction and Building Materials 73:771-780. doi: https://doi.org/10.1016/j.conbuildmat.2014.09.054

[9] M. Castelli, L. Vanneschi, S. Silva (2013), P*rediction of high performance concrete strength using Genetic Programming with geometric semantic genetic operators.* Expert Systems with Applications 40 (17):6856-6862. doi: https://doi.org/10.1016/j.eswa.2013.06.037

[10] R.E.-K. Penido, R.C.F. da Paixão, L.C.B. Costa, R.A.F. Peixoto, A.A. Cury, J.C. Mendes (2022), *Predicting the compressive strength of steelmaking slag concrete with machine learning – Considerations on developing a mix design tool.* Construction and Building Materials 341:127896. doi:https://doi.org/10.1016/j.conbuildmat.2022.127896

[11] B. Iftikhar, S.C. Alih, M. Vafaei, M.A. Elkotb, M. Shutaywi, M.F. Javed, W. Deebani, M.I. Khan, F. Aslam (2022), *Predictive modeling of compressive strength of sustainable rice husk ash concrete: Ensemble learner optimization and comparison.* Journal of Cleaner Production 348:131285. doi:https://doi.org/10.1016/j.jclepro.2022.131285

[12] P. Samui, N.-D. Hoang, V.-H. Nhu, M.-L. Nguyen, P.T.T. Ngo, D.T. Bui (2019), *A New Approach of Hybrid Bee Colony Optimized Neural Computing to Estimate the Soil Compression Coefficient for a Housing Construction Project.* Applied Sciences 9 (22):4912

[13] V.-H. Nhu, P. Samui, D. Kumar, A. Singh, N.-D. Hoang, D. Tien Bui (2019), *Advanced soft computing techniques for predicting soil compression coefficient in engineering project: a comparative study.* Engineering with Computers. doi:10.1007/s00366-019-00772-7

[14] M.-T. Cao, N.-D. Hoang, V.H. Nhu, D.T. Bui (2020), *An advanced meta-learner based on artificial electric field algorithm optimized stacking ensemble techniques for enhancing prediction accuracy of soil shear strength.* Engineering with Computers. doi:10.1007/s00366-020-01116-6

[15] D. Tien Bui, N.-D. Hoang, V.-H. Nhu (2018), *A swarm intelligence-based machine learning approach for predicting soil shear strength for road construction: a case study at Trung Luong National Expressway Project (Vietnam).* Engineering with Computers. doi:10.1007/s00366-018-0643-1

[16] H. Moayedi, M. Gör, M. Khari, L.K. Foong, M. Bahiraei, D.T. Bui (2020), *Hybridizing four wise neural-metaheuristic paradigms in predicting soil shear strength. Measurement 156:107576.* doi: https://doi.org/10.1016/j.measurement.2020.107576

[17] M.A. Shahin (2015), *A review of artificial intelligence applications in shallow foundations.* International Journal of Geotechnical Engineering 9 (1):49-60. doi:10.1179/1939787914Y.0000000058

[18] M.A. Shahin (2016), *State-of-the-art review of some artificial intelligence applications in pile foundations.* Geoscience Frontiers 7 (1):33-44. doi:https://doi.org/10.1016/j.gsf.2014.10.002

[19] R.M.L.R.F. Brasil, M.A.d. Silva (2021), *Project Optimization Using MATLAB and SOLVER.* Walter de Gruyter GmbH, Berlin/Boston

[20] H. Zhao, S. Li, B. Chen (2021), *The Reliability-Based Design Optimization of considering Rock-Support Interaction for Rock Tunnels.* Advances in Civil Engineering 2021:9921881. doi:10.1155/2021/9921881

[21] P. Goos, H. Leemans (2004), *Teaching Optimal Design of Experiments Using a Spreadsheet.* Journal of Statistics Education 12 (3):null-null. doi:10.1080/10691898.2004.11910631

[22] S.O. Haykin, Neural Networks and Learning Machines, Pearson2008.

[23] M.T. Hagan, H.B. Demuth, M.H. Beale, O.D. Jesús, Neural Network Design (2nd Edition), Martin Hagan; 2 edition (September 1, 2014), ISBN-10: 09717321162014.

[24] V.-H. Nhu, N.-D. Hoang, V.-B. Duong, H.-D. Vu, D. Tien Bui (2019), *A hybrid computational intelligence approach for predicting soil shear strength for urban housing construction: a case study at Vinhomes Imperia project, Hai Phong city (Vietnam).* Engineering with Computers. doi:10.1007/s00366-019-00718-z

[25] Y. LeCun, L. Bottou, G.B. Orr, K.-R. Müller, *Efficient BackProp,* in: G.B. Orr, K.-R. Müller (Eds.) Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 9-50.

[26] E. Momeni, R. Nazir, D. Jahed Armaghani, H. Maizir (2014), *Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN.* Measurement 57:122-131. doi: https://doi.org/10.1016/j.measurement.2014.08.007