Numerical Methods in Civil Engineering, 7-1 (2022) 09-15

Numerical Methods in Civil Engineering

Journal Homepage: https://nmce.kntu.ac.ir/

Prediction of soil permeability coefficient using the GEP approach

Mina Torabi*, Hamed Sarkardeh and S.Mohammad Mirhosseini*****

ARTICLE INFO

RESEARCH PAPER

Article history:
Received:
February 2022.
Revised:
April 2022.
Accepted:
April 2022.

Keywords:
Soft computing,
Soil infiltration,
GEP,
Coefficient of permeability,
Statistical parameters

Abstract:

Hydraulic permeability of soil (k) is a critical parameter for the mathematical modeling of groundwater and soil water flow. Due to the complexity of k, it is hard to develop a general empirical model which provides a reliable prediction of it. Therefore, this study used the gene expression programming (GEP) model as a powerful data-driven technique for estimating k. The available published data for estimation of k were culled from the literature. Six effective parameters, including clay content (CC), water content (ω), liquid limit (LL), plastic limit (PL), specific density (γ), and void ratio (e), were used to establish a predictive formula to estimate k. Statistical parameters such as bias (BIAS), root mean square error (RMSE), scatter index (SI), correlation coefficient (R), and mean absolute error (MAE) were used to evaluate the accuracy of the developed GEP model. In addition, GEP findings were compared to the artificial neural network (ANN) algorithm to assess the performance of GEP. The GEP with BIAS = −0.0005, RMSE = 0.0079, SI = 57.33%, R = 0.8109 and MAE = 0.0047 outperformed the ANN with BIAS = 0.001, RMSE = 0.0090, SI = 65.12%, R = 0.7490 and MAE = 0.0053 in predicting k in the testing stage. GEP provided an explicit mathematical equation that can be utilized to determine k. Comparing the observed data and ANN results demonstrated that the GEP approach has suitable performance for predicting k.

1. Introduction

Soil permeability is one of the most important hydraulic parameters in simulating water movement and solute transport in soil profiles, which is highly variable in space. The essential parameters in the flow and infiltration models are the hydraulic conductivity and soil water retention function k, which is one of the most critical hydraulic parameters. Determining the soil permeability coefficient is crucial, and this task is difficult, time- consuming, and expensive. The permeability is represented by the amount of water transmitted via an interconnected soil void. Soil hydraulic properties reflect the structure of the porous soil system, comprising pores of different geometry, sizes and connectivity.

For these reasons, many researchers studied k using different methods. Sihag et al. [1] assessed the potential of machine learning approaches, including support vector regression (SVR) and Gaussian process (GP) regression of cumulative infiltration, and compared their performances with three traditional models such as the Kostiakov model, US-Soil Conservation Service (SCS) model and Philip's model. Their analysis showed that the GP regression-based approach works better than the SVR, Kostiakov model, SCS model, and Philip's model approaches and could be successfully used to predict cumulative infiltration data. Sinha and Wang [2] presented artificial neural network (ANN) prediction models which relate permeability, maximum dry density (MDD), and optimum moisture content with the classification properties of soils. Chapuis [3] proposed a new equation based on a best-fit equation in a graph of the logarithm of measured k. Boadu [4] predicted k values using grain size distribution, fractal dimension, distribution entropy, porosity, soil density, and fine grain content. Elbisy [5] developed two regression equations to

*Ph.D. Candidate, Department of Civil Engineering, Arak Branch, Islamic Azad University, Arak, Iran. Email: mina_torabi89@yahoo.com

** Associate Professor, Department of Civil Engineering, Faculty of Engineering, Hakim Sabzevari University, Sabzevar, Iran. Email: sarkardeh@hsu.ac.ir

*** Corresponding Author: Assistant Professor, Department of Civil Engineering, Arak Branch, Islamic Azad University, Arak, Iran. Email: m-mirhoseini@iau-arak.ac.ir

[DOR: 20.1001.1.23454296.2022.7.1.6.0]

[DOI: 10.52547/nmce.2022.414]

estimate the field saturated hydraulic conductivity *k* values for a soil sample from its properties. The SVM approach was proposed and adopted to forecast the k field based on soil properties easily measured in the laboratory, and a genetic algorithm (GA) was used to optimize the parameters of the SVM. Zhu et al. [6] conducted a study to detect the differences and vertical variations of k to identify its dominant influencing factors. The results are helpful for understanding water movement within the soil profile and developing hydrological models.

Recently, data-driven methods in civil engineering have become excellent tools for analyzing various civil engineering challenges. The GEP and ANN methods are frequently utilized in various civil engineering fields, such as estimating scour depth [7,8], longitudinal dispersion coefficient [9], water quality index [10], dynamic pressure distribution in hydraulic structures [11,12], and rock quality index [13].

The ANN technique is a predictive modeling tool with input and output variables described by coefficient weights between neurons. The primary shortcoming of neural networks is the inherent black-box nature. As a result, it may not be easy to express the relationship between the inputs and outputs of a neural network in mathematical expression form [14]. A newer computational intelligence approach known as gene expression programming (GEP), which belongs to a group of evolutionary programming techniques, is also a potential option for handling complicated estimation issues. GEP is a white-box model that can generate equations to estimate an output parameter based on the influential input variables and presents a clear relationship between input and output variables. In contrast, an ANN model is a "black-box" model that does not show how input and output variables are linked and is difficult to use.

Due to its high predictive accuracy, GEP is commonly employed in the literature to conduct predictive modeling of engineering problems. GEP can generate a predictive model even without a predefined equation, like regression analysis. Therefore, this study employed the GEP to address the challenge of predicting the soil permeability coefficient. GEP was utilized to evaluate its feasibility as a predictive tool for the computation of k. A GEP-based equation was proposed for the estimation of k. The estimation of k was then solved using ANN, and the resulting ANN and GEP solutions were compared with others and observed values for assessing the performance of GEP. ANN modeling was also used to understand GEP's prediction performance. In predicting the soil permeability coefficient, six parameters, namely, CC, ω, LL, PL, γ, e were used. As mentioned before, GEP is a white-box data-driven model that provides a mathematical expression for predicting *k*. Thus, in this study, *k* was predicted using field data and soil physical properties by GEP.

2. Materials and Methods

2.1 Theoretical background and availability of data set

In functional form, the permeability of soil is [15]:

$$k = f\,(CC,\ \omega,\ LL,\ PL,\ \gamma,\ e) \tag{1}$$

Eighty-four data samples are available for modeling *k*. For more details about the data set found in Ref. [15], Table 1 summarizes the major statistical parameter of the data used in the present study [15].

Table 1: Basic statistical parameters of data used in this study [14]

Variable	Min	Max	Average
$CC(\%)$	5.70	64.00	25.17
$\omega(\%)$	15.09	99.90	34.23
$LL(\%)$	18.90	88.93	37.27
$PL(\%)$	12.20	54.80	22.21
$\gamma(\frac{gr}{cm^3})$	2.58	2.74	2.68
e	0.46	2.63	0.97
$k(10^{-9}cm/s)$	0.003	0.071	0.015

2.2 Overview of GEP and development

Ferreira [16] developed GEP, a novel evolutionary approach for artificial intelligence. This approach is a refinement of GP introduced by Koza. GA, GP, and GEP are three algorithms classified as genetic algorithms since they all employ populations of individuals, select individuals based on their fitness, and add genetic diversity through one or more genetic operators. The primary distinction between the three methods is in the nature of the individuals: in GAs, individuals are represented by symbolic strings of constant length (chromosomes); in GP, individuals are represented by nonlinear entities with varying dimensions and shapes (parse trees); and in GEP, individuals are also represented by nonlinear entities with varying dimensions and shapes (expression trees); however, these complex entities are encrypted as simple strings of constant length. Most genetic operators used in GAs may also be applied with modest modifications in GP and GEP [17-19]. GEP development consists of five major parts: "fitness function," "terminal set," "function set," "control components," and "stop criterion." These major parts must be assigned for solving a problem using GEP.

GEP fitness functions include *RMSE*, *MAE*, and root-relative square error (RRSE) examined to generate the GEP

[DOR: 20.1001.1.23454296.2022.7.1.6.0]
[DOI: 10.52547/nmce.2022.414]

equation. The RRSE fitness function was found to have better performance than *RMSE* and *MAE*. In addition, several previous studies reported successful application of RRSE for GEP development.

The next major stage is to define the terminal and function sets used to produce chromosomes. As a result, the terminal set for k prediction is input variables including CC, ω, LL, PL; γ, and e. Additionally, the following function sets were used: $+, -, \times, \div, \sqrt{x}, \sqrt[3]{x}, x^2, x^3, exp, ln, Sin, Cos, Atan$.

Then, the architecture of chromosomes, which includes the number of genes and head length, was defined. In addition, the summation operator was selected as the linking function so that summation of sub-expression trees and the genetic operator's value are determined for the GEP model (See Table 2). The stop criterion for GEP was select 100000 via trial and error procedure.

Table 2: Setting parameters of GEP

Parameter	Value
Number of chromosomes	50
Number of genes	4
Head size	8
One-point recombination rate	0.3
Two-point recombination rate	0.3
Gene recombination	0.1
Gene transposition	0.1
Inversion rate	0.1
Mutation rate	0.044

The GEP presents four Sub-ETs for estimation of k. The Sub-ETs obtained via GEP are illustrated in Fig 1.

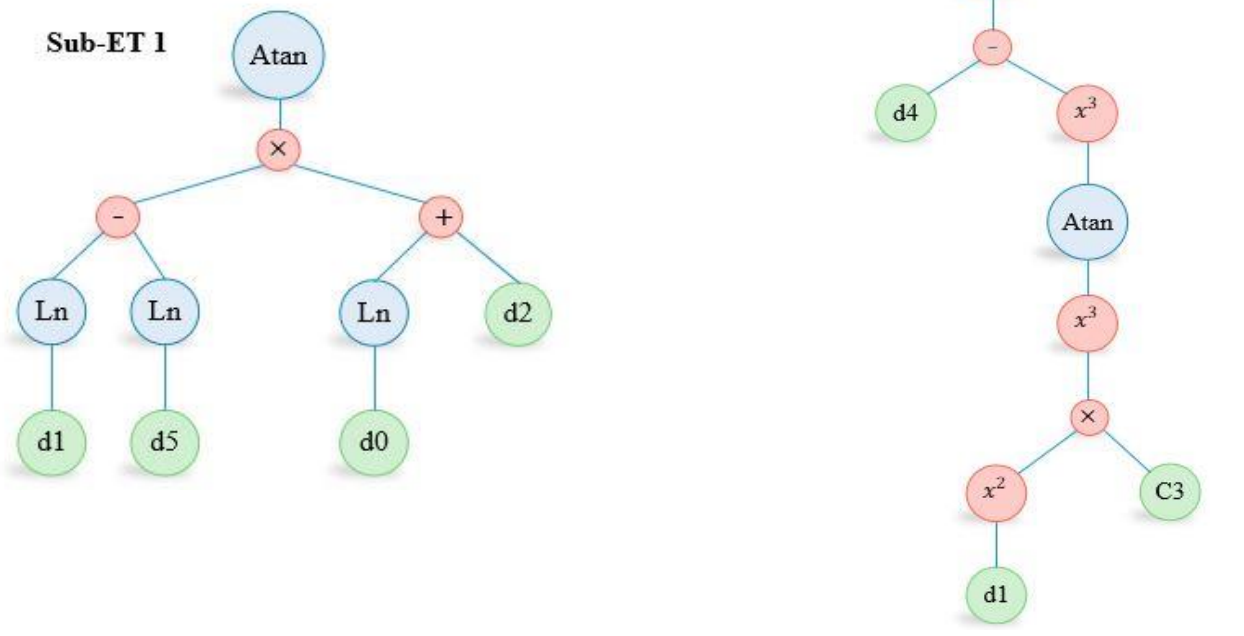

Fig. 1: Sub-ETs for prediction of k

[DOR: 20.1001.1.23454296.2022.7.1.6.0]
[DOI: 10.52547/nmce.2022.414]

The explicit equation is related to each Sub-ETs indicated in Figure 1 as follows:

Sub-ET 1: $Arctan((\ln(d_1) - \ln(d_5)) \times (\ln(d_0) + d_2))$ (2)

Sub-ET 2: $d_0 \times (((d_3 - d_2) \times (C1/d_4)) - d_4)$ (3)

Sub-ET 3: $Arctan(d_0^2 \times (C2 \times (\sqrt[3]{d_3}/d_4)))$ (4)

Sub-ET 4: $Arctan(d_4 - Arctan(d_4 - \left(Arctan((d_1^2 \times C3)^3)\right)^3))$ (5)

The variables of Sub-ETs were $d_0 = CC$, $d_1 = e$, $d_2 = \gamma$; $d_3 = LL$,$d_4 = \omega$; $d_5 = PL$.

In addition, the constant values of each Sub-ETs were C1 = 9.748, C2 = 3.972, and C3 = 2.711.

Finally, the value of k was calculated by summation of sub-ETs as follows:

$k = Sub - ET\ 1 + Sub - ET\ 2 + Sub - ET\ 3 + Sub - ET\ 4$ (6)

As can be seen, an explicit mathematical equation was obtained by GEP. According to the properties of this algorithm, triangular and algebraic functions have been used to estimate k using input variables. This led to the flexibility and increase of computational capabilities of GEP for computation of k.

2.3 Overview of ANN and development

Neural networks are constructed from a collection of simple parts called neurons that work in parallel. These components are inspired by the nervous system of the human brain. As is the case in nature, the function of a network is primarily determined by the connections between its neurons. A neural network can be trained to serve a specific purpose by changing the values of the weights between the neurons in the network.

ANN is composed of multiple layers. Each layer has one or more neurons that depend on the number of input data. Each neuron in the first layer takes the input data, multiplies it by the connection weight, and sends the result to the corresponding neurons in the hidden layer, which applies the activation function. The hidden layer's results are then sent to the output layer by multiplying the output of each neuron in the hidden layer by the connection weight between the hidden and output neurons. Finally, the output layer presents the results of the network. Afterward, the error is calculated by comparing the output values to the target output at this phase.

If the error is within reasonable limits, the result of the network is valid; otherwise, the connections' weights are modified beginning at the output layer and propagating backward. After updating the weights, a new cycle starts until training is completed. The training procedure is terminated after reaching a certain error level or completing a certain number of cycles. Numerous publications have explained the fundamentals of artificial neural networks [20]. The dataset presented in Sect. 2.1 was also employed in the ANN modeling of *k*. For ANN model development, similar GEP and the same input and output variables were used. In addition, for the assessment of GEP, the same train data set used for GEP was employed for the training of ANN. The primary objective of training ANN is to establish the optimal weights between neurons in the configuration ANN for predicting *k*. This research applies a multilayer feed-forward neural network with a back-propagation training algorithm to train ANNs, a frequently accepted technique. Based on Hornik et al.'s [21] study, one hidden layer was employed for ANN development. The learning rate and momentum coefficient were set to 0.25 and 0.9 based on Swingler's [22] recommendation. In addition, the number of hidden neurons was obtained via Huang and Foo's criterion (2002), successfully used in Refs. [23,24], which is as follows:

$N^H < 2N^I + 1$ (7)

where N^H and N^I are the number of hidden and input neurons in the hidden and input layers, respectively. The best configuration of ANN was found via trial and error. The most accurate results were obtained using a multilayer perceptron (MLP) network with one hidden layer and six neurons in the hidden layer. Therefore, the best ANN configuration was 6×6×1. This configuration is displayed in Figure 2.

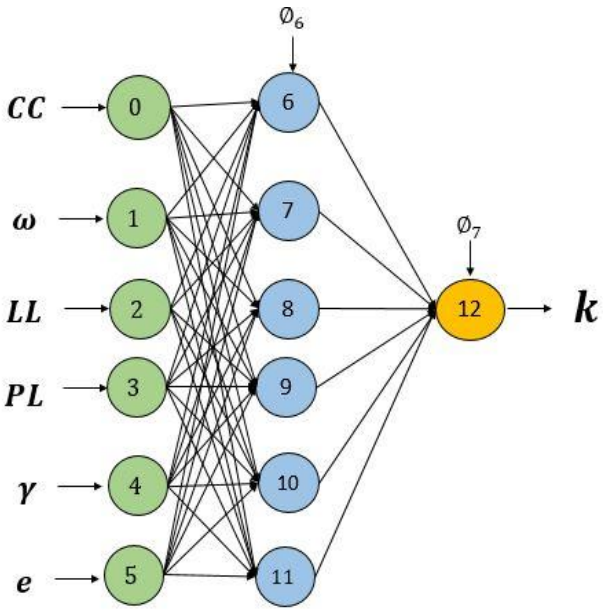

Fig. 2: The ANN configuration for prediction *k*

The optimal weights and bias values were also extracted from the proposed ANN for future work. Tables 3 and 4 present the corresponding optimal weights between input, hidden, and output layers.

[DOR: 20.1001.1.23454296.2022.7.1.6.0]
[DOI: 10.52547/nmce.2022.414]

3. Performance evaluation criteria

For evaluating the accuracy and ability of developed models, the common statistical criteria include bias (*BIAS*), root mean square error (*RMSE*), scatter index (*SI*), correlation coefficient (*R*), and mean absolute error (*MAE*) were used. Accordingly, the following equations were used to calculate the efficacy of the suggested models [25,26]:

$$BIAS = \overline{k_p} - \overline{k_o} \quad (8)$$

$$RMSE = \sqrt{\sum_{i=1}^{N} \left(k_{io} - k_{ip}\right)^2 / N} \quad (9)$$

$$SI = \frac{RMSE}{\overline{k_o}} \times 100 \quad (10)$$

$$R = \frac{\sum_{i=1}^{n} \left(k_{io} - \overline{k_o}\right)\left(k_{ip} - \overline{k_p}\right)}{\sqrt{\sum_{i=1}^{n} \left(k_{io} - \overline{k_o}\right)^2} \sqrt{\sum_{i=1}^{n} \left(k_{ip} - \overline{k_p}\right)^2}} \quad (11)$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left|k_{io} - k_{ip}\right| \quad (12)$$

Here k_{io} and k_{ip} represent the observed and predicted values, $\overline{k_o}$ and $\overline{k_p}$ represent the average of the observed and predicted values of k_{io} and k_{ip} , respectively; and N equals the number of the dataset.

Table 3: The optimal weights between input and hidden layers

Weight between input and hidden neurons						Bias
$w_{06} = -0.62$	$w_{07} = -0.22$	$w_{08} = 0.04$	$w_{09} = -0.11$	$w_{010} = 0.25$	$w_{011} = 0.02$	$\emptyset_6 = -0.33$
$w_{16} = -0.56$	$w_{17} = -0.71$	$w_{18} = -1.19$	$w_{19} = 0.04$	$w_{110} = -0.19$	$w_{111} = -1.18$	$\emptyset_7 = 0.06$
$w_{26} = -0.14$	$w_{27} = -0.26$	$w_{28} = -0.32$	$w_{29} = -0.18$	$w_{210} = -0.24$	$w_{211} = -0.22$	$\emptyset_8 = 0.34$
$w_{36} = -0.33$	$w_{37} = 0.14$	$w_{38} = -1.05$	$w_{39} = 0.25$	$w_{310} = 0.20$	$w_{311} = -1.09$	$\emptyset_9 = 0.11$
$w_{46} = 0.63$	$w_{47} = 0.28$	$w_{48} = 0.92$	$w_{49} = -0.10$	$w_{410} = 0.10$	$w_{411} = 0.99$	$\emptyset_{10} = -0.58$
$w_{56} = -0.27$	$w_{57} = -0.32$	$w_{58} = -0.29$	$w_{59} = -0.56$	$w_{510} = -0.02$	$w_{511} = -0.96$	$\emptyset_{11} = 0.65$

Table 4: The optimal weights between hidden and output layers

Weight between hidden and output neurons						Bias
$w_{612} = -0.83$	$w_{712} = -0.38$	$w_{812} = -1.58$	$w_{912} = 0.09$	$w_{1012} = 0.24$	$w_{1112} = -2.09$	$\emptyset_{11} = 0.73$

RMSE and *MAE* have the benefit of calculating model error in just the same unit of variables. *R* provides a metric for the model's ability to mimic observed results. One of the primary advantages of *R* and *SI* are their non-dimensionality, which enables the evaluation of various models irrespective of the size and dimension of the variables. The *BIAS* values indicated underestimating and overestimating results of GEP and ANN models. The negative and positive values of *BIAS* indicate judgments about under and overestimated results obtained from the models, respectively.

Table 5 shows the values of statistical parameters for ANN and GEP in the training and testing datasets to predict *k*. Table 5 presents the statical metrics of GEP and ANN for predicting *k*. GEP in training stage had *BIAS* = −0.0006, *RMSE* = 0.0069, *SI* = 44.76 and *MAE* = 0.0041 and ANN has *BIAS* = 0.0002, *RMSE* = 0.0075, *SI* = 48.70%, *R* = 0.8986 and *MAE* = 0.0045. As seen, the error values are In addition, the *R*-value increased 8.26% when using GEP compared to ANN.

The *BIAS* values obtained for GEP and ANN are almost equal to zero. However, the negative *BIAS* value obtained through GEP indicated a slight underestimation of the *k* values. In contrast, the positive *BIAS* value obtained via ANN indicates a marginal overestimation of the *k* values. Scatter plots of ANN and GEP are displayed in Figures 3 and 4. These scatter plots graphically indicate the performance of GEP and ANN. As seen, these figures found similar performances of ANN and GEP for the prediction of *k*. However, GEP results are more concentrated on the 45-degree line.

In addition, to further evaluate the results of this study, they were compared with previous work conducted by Pham et al. [15]. They used the M5 model tree (M5MT) and Gaussian process (GP) approach to predict *k*. They discovered that M5MT, with *RMSE* = 0.0081 and *MAE* = 0.0045, outperforms GP, with *RMSE* = 0.0093 and *MAE* = 0.0054. In the present study, GEP with *RMSE* = 0.0079 and *MAE* = 0.0047 led to similar results to M5MT. In addition, compared to previous work [15], this study provided an explicit equation for predicting *k*.

[DOR: 20.1001.1.23454296.2022.7.1.6.0]
[DOI: 10.52547/nmce.2022.414]

[DOR: 20.1001.1.23454296.2022.7.1.6.0]
[DOI: 10.52547/nmce.2022.414]

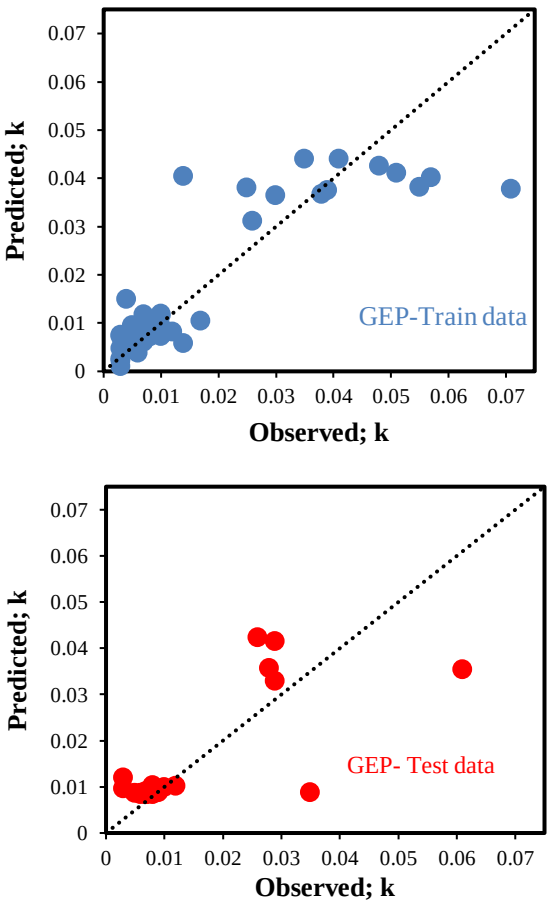

Fig. 3: Observed data versus the output of GEP for a) training and b) testing data set.

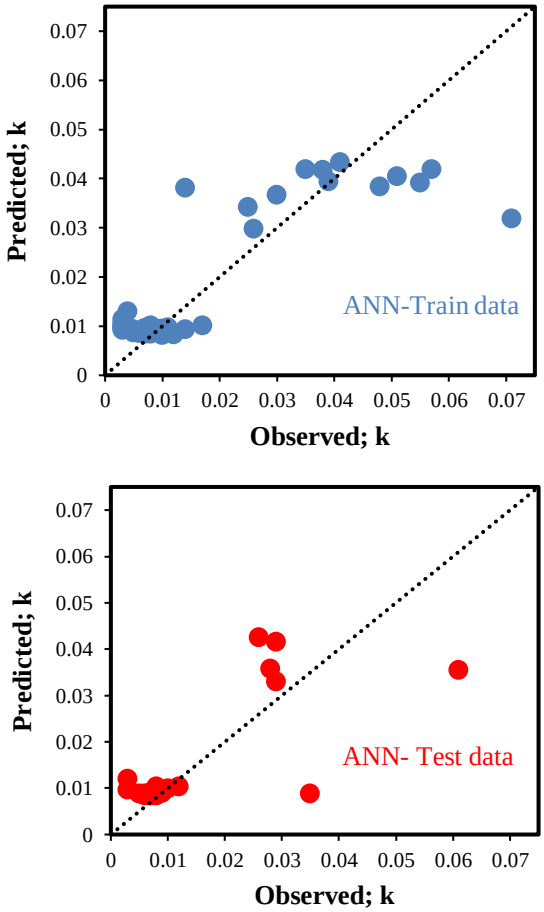

Fig. 4: Observed data versus the output of ANN for a) training and b) testing data set

Table 5: Statistical metric values of GEP and ANN for prediction *k*

Model	*BIAS*	*RMSE*	*SI* (%)	*R*	*MAE*
GEP (Train)	−0.0006	0.0069	44.76	0.9115	0.0041
GEP(Test)	−0.0005	0.0079	57.33	0.8109	0.0047
ANN(Train)	0.0002	0.0075	48.70	0.8986	0.0045
ANN (Test)	0.0010	0.0090	65.12	0.7490	0.0053

4. Summary and Conclusions

In this study, GEP was applied to solve the problem of soil permeability coefficient. This study provided a mathematical expression obtained by GEP for predicting *k*. These equations calculate the *k* value by substituting the input variables. The data samples were obtained from previously published research. Moreover, the ANN model was employed to evaluate the potential of GEP in predicting k. Six input parameters, including clay content (CC), natural water content (ω), liquid limit (LL), plastic limit (PL), specific density (γ), and void ratio (e) were used to estimate the soil permeability coefficient (k). Statistical parameters were used to determine the performance of the GEP equation. The GEP in the training stage with *RMSE* = 0.0069 and R = 0.9115 led to a slightly better prediction of *k* compared to ANN with *RMSE* = 0.0075 and R = 0.8986. Compared to ANN with *RMSE* = 0.009 and R = 0.7490, GEP with *RMSE* = 0.0079 and R = 0.8109 had a better performance in estimation of *k* in the testing stage. Therefore, the GEP results were much better than the best ANN outputs. Nonetheless, GEP provided an explicit expression for prediction *k*. Thus, from this point of view, GEP is preferable to ANN. However, ANN presented a complex network with optimal weights for predicting k. Overall, the findings of the present research indicated that GEP outperformed ANN. For future work, the application of the group method of data handling (GMDH), extreme learning machine (ELM), and evolutionary polynomial regression (EPR) is suggested for the prediction of *k*.

References

[1] Sihag, P., Tiwari, N. K., & Ranjan, S. (2017). Modelling of Infiltration of Sandy Soil Using Gaussian Process Regression. Modeling Earth Systems and Environment, 3(3), 1091-1100.

[2] Sinha, S. K., & Wang, M. C. (2008). Artificial Neural Network Prediction Models for Soil Compaction and Permeability. Geotechnical and Geological Engineering, 26(1), 47-64.

[3] Chapuis, R. P. (2004). Predicting the Saturated Hydraulic Conductivity of Sand and Gravel Using Effective Diameter and Void Ratio. Canadian Geotechnical Journal, 41(5), 787-795.

[4] Boadu, F. K. (2000). Hydraulic Conductivity of Soils From Grain-size Distribution: New Models. Journal of Geotechnical and Geoenvironmental Engineering, 126(8), 739-746.

[5] Elbisy, M. S. (2015). Support Vector Machine and Regression Analysis to Predict the Field Hydraulic Conductivity of Sandy Soil. KSCE Journal of Civil Engineering, 19(7), 2307-2316.

[6] Zhu, P., Zhang, G., & Zhang, B. (2022). Soil Saturated Hydraulic Conductivity of Typical Revegetated Plants on Steep Gully Slopes of Chinese Loess Plateau. Geoderma, 412, Article e115717.

[7] Samadi, M., Afshar, M. H., Jabbari, E., & Sarkardeh, H. (2021). Prediction of Current-induced Scour Depth Around Pile Groups Using MARS, CART, and ANN Approaches. Marine Georesources & Geotechnology, 39(5), 577-588.

[8] Samadi, M., Jabbari, E., Azamathulla, H. M., & Mojallal, M. (2015). Estimation of Scour Depth Below Free Overfall Spillways Using Multivariate Adaptive Regression Splines and Artificial Neural Networks. Engineering Applications of Computational Fluid Mechanics, 9(1), 291-300.

[9] Alizadeh, M. J., Shabani, A., & Kavianpour, M. R. (2017). Predicting Longitudinal Dispersion Coefficient Using ANN with Metaheuristic Training Algorithms. International Journal of Environmental Science and Technology, 14(11), 2399-2410.

[10] Mojaradi, B., ALIZADEH, S. F., & Samadi, M. (2018). Estimation of Water Quality Index in Talar River Using Gene Expression Programming and Artificial Neural Networks. Iranian Journal of Watershed Management Science and Engineering, 12(41), 61- 72.

[11] Samadi, M., Sarkardeh, H., & Jabbari, E. (2021). Prediction of the Dynamic Pressure Distribution in Hydraulic Structures Using Soft Computing Methods. Soft Computing, 25(5), 3873-3888.

[12] Samadi, M., Sarkardeh, H., & Jabbari, E. (2020). Explicit Data-driven Models for Prediction of Pressure Fluctuations Occur During Turbulent Flows on Sloping Channels. Stochastic Environmental Research and Risk Assessment, 34(5), 691-707.

[13] Taban, M. H., Hajiazizi, M., & Ghobadian, R. (2019). Predicting the Value of the Rock Quality Index in the Q-system Using Gene Expression Programming. Journal of Numerical Methods in Civil Engineering, 4(2), 44-54.

[14] Samadi, M., Afshar, M. H., Jabbari, E., & Sarkardeh, H. (2020). Application of Multivariate Adaptive Regression Splines and Classification and Regression Trees to Estimate Wave-induced Scour Depth Around Pile Groups. Iranian Journal of Science and Technology, Transactions of Civil Engineering, 44(1), 447-459.

[15] Pham, B.T., Ly, H.B., Al-Ansari, N. and Ho, L.S., 2021. A Comparison of Gaussian Process and M5P for Prediction of Soil Permeability Coefficient. Scientific Programming, 2021, Article e3625289.

[16] Ferreira, C. (2006). Gene Expression Programming. Berlin: Springer.

[17] Terzi, Ö., & Ergin, G. (2014). Forecasting of Monthly River Flow with Autoregressive Modeling and Data-driven Techniques. Neural Computing and Applications, 25(1), 179-188.

[18] Karbasi, M., & Azamathulla, H. M. (2017). Prediction of Scour Caused by 2D Horizontal Jets Using Soft Computing Techniques. Ain Shams Engineering Journal, 8(4), 559-570.

[19] Terzi, Ö. (2013). Daily Pan Evaporation Estimation Using Gene Expression Programming and Adaptive Neural-based Fuzzy Inference System. Neural Computing and Applications, 23(3), 1035-1044.

[20] Haykin, S. (1994). Neural networks, 2nd Edition. New York: Prentice Hall.

[21] Hornik, K., Stinchombe, M., & White, H. (1989). Multilayer Feed-forward Network and Universal Approximator. Neural Network. 2(5), 359-366.

[22] Swingler, K. (1996). Applying Neural Networks: A Practical Guide. New York: Academic Press.

[23] Ayoubloo, M. K., Azamathulla, H. M., Jabbari, E., & Zanganeh, M. (2011). Predictive Model-based for the Critical Submergence of Horizontal Intakes in Open Channel Flows with Different Clearance Bottoms Using CART, ANN and Linear Regression Approaches. Expert Systems with Applications, 38(8), 10114-10123.

[24] Kamranzad, B., & Samadi, M. (2013). Assessment of Soft Computing Models to Estimate Wave Heights in Anzali Port. Journal Of Marine Engineering, 9(17), 27-36.

[25] Samadi, M., Jabbari, E., & Azamathulla, H. M. (2014). Assessment of M5′ Model Tree and Classification and Regression Trees for Prediction of Scour Depth Below Free Overfall Spillways. Neural Computing and Applications, 24(2), 357-366.

[26] Samadi, M., & Jabbar, E. (2012). Assessment of Regression Trees and Multivariate Adaptive Regression Splines for Prediction of Scour Depth Below the Ski-Jump Bucket Spillway. Journal of Hydraulics, 7(3), 73-79.

[DOR: 20.1001.1.23454296.2022.7.1.6.0]

[DOI: 10.52547/nmce.2022.414]