Forecasting Factor of Safety of Slopes Stability Using Several Machine Learning Techniques

Arsalan Mahmoodzadeh (✉ arsalan.mahmoodzadeh@uoh.edu.iq)

University of Halabja

Mokhtar Mohammadi

Lebanese French University

Research Article

Keywords: Slope stability, Factor on safety, Machine learning, PLAXIS, Feature selection.

Posted Date: August 4th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-669928/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Forecasting Factor of Safety of Slopes Stability using Several Machine Learning Techniques

^{a,1}Arsalan Mahmoodzadeh*, ^{b,2}Mokhtar Mohammadi

^a Department of Building and Construction Engineering, University of Halabja, Halabja, Kurdistan Region, Iraq

^b Department of Information Technology, Lebanese French University, Erbil, Iraq

¹Email: arsalan.mahmoodzadeh@uoh.edu.iq

²Email: mukhtar@lfu.edu.krd

*Corresponding author (A. Mahmoodzadeh)

Abstract

Because of the disasters associated with slope failure, the analysis and forecasting of slope stability for geotechnical engineers are crucial. In this work, in order to forecast the factor of safety (FOS) of the slopes, six machine learning (ML) techniques of Gaussian process regression (GPR), support vector regression (SVR), decision trees (DT), long-short term memory (LSTM), deep neural networks (DNN), and K-nearest neighbors (KNN) were performed. A total of 327 slope cases in Iran with various geometric and shear strength parameters analyzed by PLAXIS software to evaluate their FOS, were employed in the models. The K-fold (K=5) cross-validation (CV) method was applied to evaluate the performance of models' prediction. Finally, all the models produced acceptable results and almost close to each other. However, the GPR model with $R^2 = 0.8139$, RMSE = 0.160893, and MAPE = 7.209772%, was the most accurate model to predict slope stability. Also, the backward selection method was applied to evaluate the contribution of each parameter in the prediction problem. The results showed that all the features considered in this study have significant contributions to slope stability. However, features ϕ (friction angle) and γ (unit weight) were the most effective and least effective parameters on slope stability, respectively.

Keywords: Slope stability, Factor on safety, Machine learning, PLAXIS, Feature selection.

1. Introduction

In geotechnical engineering, the analysis of slope stability is of importance. Economics has developed over time and, with that progress, the number of slopes is growing. As a result, landslides affected by slope instability have been among the three most catastrophic natural disasters in the world, besides volcanoes and earthquakes. In order to minimize or avoid landslide failure, it is significant to understand the importance of slope stability analysis. It is complicated to anticipate the stability of slopes due to the complexity of them because they are not dependent on specified data or a single parameter, but they are mostly dependent on a variety of variables, such as slope angle, slope height, soil properties, and seasonally changing soil water content (Sakellariou and Ferentinou 2005; Duncan 2000; Suman et al. 2016).

While there is still much to be done to predict and evaluate slope stability, considerable research has been performed across several parts of the world. Several useful models have been implemented in the previous studies by different researchers to analyze and predict slopes' failures (Lin et al. 2018). These approaches can be categorized into four groups including qualitative analysis (Bye and Bell 2001; Trivedi et al. 2012), quantitative analysis (Thiebes et al. 2013; Thiebes et al. 2014), physical simulation (He et al. 2012; Kim et al. 2016; Sarkar et al. 2012), and field monitoring analysis (Pirone et al. 2015). These researchers have done a remarkable job of tackling slope stability. However, the

38 problem is that the techniques they built are not optimal and only give specific solutions for certain slopes. These
39 models are not suitable to solve all slope stability problems completely because of several drawbacks (Lin et al. 2018):

40 a) The methods used are not really appropriate because of the shortcomings. For instance, the limit equilibrium
41 method cannot consider the constitutive relationship of soil, and it needs to search for the most dangerous sliding
42 surface. Besides, the analysis is complicated for multi-layer soils. Therefore, this approach does not accurately
43 elucidate the proper safety and reliability during the stability slopes analysis.

44 b) The safety factor is not clearly defined. In general, slopes with a factor of safety (FOS) of greater than 1.20 are safe.
45 In spite of this, there have been failed slopes with the FOS greater than 1.2 in real engineering.

46 c) These methods are affected by human subjective factors.

47 In recent years, artificial intelligence and soft computing models have been broadly applied in geotechnical
48 applications, e.g., tunnel resources (Mahmoodzadeh and Zare 2016; Mahmoodzadeh et al. 2019; Mahmoodzadeh et
49 al. 2020a; Mahmoodzadeh et al. 2020b; Mahmoodzadeh et al. 2021a), mechanical property (Liu et al. 2013; Liu et al.
50 2015), tunnel surface settlement (Mahmoodzadeh et al. 2020c), landslide displacement (Youssef et al. 2016), rock
51 bursts (Zhou et al. 2016), tunnel geomechanical properties (Mahmoodzadeh et al. 2021b; Mahmoodzadeh et al.
52 2021c), sidewall displacement of underground caverns (Mahmoodzadeh et al. 2021d).

53 In the meantime, several supervised learning approaches have recently been available for slope stability
54 computation. These methods have shown remarkable results (Lin et al., 2018). For instance, Lu and Rosenbaum (2003)
55 were able to merge artificial neural networks and the gray system approach for analyzing the stability of slopes based
56 on the acquired situations' geotechnical characteristics and historical behavior. Artificial neural networks (ANN) were
57 implemented by Wang et al. (2005) and Das et al. (2011) in order for the stability analyses of slopes. Zhao et al. (2012)
58 investigated the correlation between slope stability and its effect parameters using the relevance vector machine. Liu
59 et al. (2014) evaluated 97 cases using the extreme learning machine method to assess slope stability. Furthermore,
60 through 168 situations, Hoang and Pham (2016) have implemented metaheuristic optimize least squares support
61 vector regression for slope analysis.

62 Manouchehrian et al. (2014) developed a genetic algorithm to attempt to anticipate the FOS for slopes and have
63 shown that this model is reasonably straightforward and efficient. Xue et al. (2014) suggested particle swarm
64 optimization (PSO) for picking the necessary factors for the support vector machines (SVM) model. The outcomes
65 revealed that the combination of SVM and PSO models is an effective analysis method for predicting slopes' stability.
66 Li et al. (2013) and Samui (2008) similarly evaluated slopes' stability based on SVM and achieved valuable outcomes. In
67 addition, Li et al. (2013) and Samui (2008) have projected slope stability using SVM and gained great outcomes. Wen
68 and Zhang (2014) gathered data from various situations, and 30 of them utilized the testing dataset for training a
69 Random Forest (RF) model that predicts slope stability. The remaining 12 data sets were used to assess the qualified
70 prediction models' efficiency and noticed that testing the model enhanced its efficiency. Khajehzadeh et al. (2012) and
71 Raihan et al. (2013) used a gravitational search algorithm (GSA) to solve global optimization problems and minimize
72 the FOS in the study of slope stability. Shi et al. (2010) trained the Bayes model with the gathered data of slope
73 situations and observed that the estimation outcomes were precisely the same as the real situations. The model may
74 be used to accurately evaluate slope stability, achieving high predictive performance and low misclassification with a
75 low error rate.

76 The mentioned intelligent models explain slopes' failures, but they are not adequate to solve them entirely. A
77 particular technique could be suitable for some situations but not appropriate for others, and no distinction between
78 intelligent modeling approaches has been observed in the prediction of slope stabilities. Therefore, the models'

credibility and precision are not well established. Furthermore, cross-validation (CV) is an efficient and accurate approach to confirm the models' prediction performance (Lin et al. 2018). It is crucial to recommend CV for accuracy verification of optimized models.

In this work, six machine learning (ML) methods were used to forecast the slopes' FOS: Gaussian process regression (GPR), support vector regression (SVR), decision trees (DT), long-short term memory (LSTM), deep neural networks (DNN), and K-nearest neighbors (KNN). PLAXIS software is used to measure the FOS of 327 slope cases in Iran, using input parameters such as unit weight (γ), cohesion (c), friction angle (ϕ), slope angle (α), slope height (H), and pore pressure ratio (r_u). These datasets are employed in the prediction models. The prediction output of the models is evaluated using the K-fold ($K=5$) CV process.

The analyzed FOSs using PLAXIS software are compared to the forecasting values made by the ML models. Finally, the best prediction model has been specified. Also, in order to show the effect of each of the input parameters on the SOF of a slope, the backward selection method is used. In this way, the most important parameters affecting the stability of slopes are identified. Fig. 1 presents the complete flowchart of the analysis.

2. Database and statistical analysis

In Fig. 2, a slope's geometry and the effective parameters on slopes' stability are shown. It should be noted that, in this study, the external factors generating the failure, such as earthquake effects and other human-made factors, have not been considered.

In Table 1, various studies were performed to predict the circular mode of slope stability utilizing soft computational methods are presented. In most of these studies, six factors of γ , c , ϕ , α , H , and r_u have been considered effective slope stability factors. Therefore, according to the literature and data availability, these six factors are considered in this study.

In order to evaluate the efficiency of the prediction models applied in this work for slope stability, a database including 344 slope cases analyzed for circular failure mechanisms was employed. Circular failure is a prevalent type of failure of slopes in soil and more fractured rock areas. To determine the value of FOS, all the slopes were studied using the finite element software of PLAXIS. These analytical outcomes are compared with the prediction results of the ML models.

In this study, data analyses of bivariate correlation and basic descriptive statics are used so that a deeper understanding of the database employed can be achieved. From the statistical point of view, when the training dataset factors are in larger variance ranges, a more applied predictive model can be developed. Also, in the developed model, the dependent variable(s) are generally predicted by interpolating the dataset. Thus, it is crucial to define an effective range of variance for the independent variable(s). Hence, to define the variance set of the parameters, simple descriptive statistical analysis is performed on the database (Manouchehrian et al. 2014). Such a common statistical tool as boxplot is used for presenting groups of datasets via their five-number summaries containing: the lower whisker, lower quartile, median, upper quartile, and upper whisker. This boxplot can also detect any available outlier in the dataset. For each parameter in the initial dataset, the boxplots are presented in Fig. 3. For each correlation, the tops of the box (25 percentage) and the bottoms of the box (75 percentage) are elucidated. There are interquartile ranges which are the spaces between the bottoms and tops. The sample median is the line that runs through the center of each box. For three boxplots of γ and c , the median is roughly in the middle of the box, while the others are not, indicating sample skewness.

The whiskers are defined by the continuation of lines above and below the boxes. The whiskers are spread from the inter-quartile range's ends to the whisker range's farthest data. Referring to Fig. 3, it can be observed that the length

linking the higher whisker and the higher quartile differs from the distance between the lower whisker and the lower quartile for H , c , and r_u factors; however, these distances are roughly the same for γ , ϕ , and α factors. In addition, it also illustrated the allocation of the data. Beyond the whisker length, there are outliers, which are the other observations. Several outlying findings are presented in the H and c parameters, as depicted in Fig. 3. To minimize the chance of misleading for statistical interpretations derived from datasets including outliers, for much more investigations, 55 outliers need to be deducted from the dataset. The medians of the c and the γ are around 10 kN/m^2 and 20 kN/m^3 , respectively. As per the study of Liu et al. (2014), Zhang et al. (2014), and Lin et al. (2018), It could be acquired that these slopes are mostly soil slopes or heavily fractured rock. The very large values of cohesion, which are outlier cases, are perhaps for rock slopes. The outcome of Zhang et al. (2014) study explained that the improvements in models' performance could be achieved with the removal of these cases. Therefore, these outliers (17 slope cases) should be eliminated, and 327 slope cases remained. Table 2 shows the outcomes of the simple descriptive statistical analysis achieved of the reducing datasets.

For the 327 datasets considered in this study, there are two slope stability modes of failure (168 cases) and stable (159 cases). The two modes of slope stability cases employed in this research are relatively balanced, as shown in Fig. 4.

In the soft computing problems, to assess their general efficiency, prediction models' performance is necessary to be verified on new datasets. Therefore, the entire dataset could be subdivided into two training and testing sets. The prediction model is trained, and the hyper-parameters are tuned using the training datasets. The testing datasets are utilized separately to measure the overall efficiency of the prediction of models.

This article applies the K-fold CV ($K=5$) method to classify the database into two groups: training and testing. The work method is such that whole datasets are divided into five equal categories so that each time one category is used as training sets, and the other four categories are used as testing sets. In the K-fold CV method, all data will be experienced in both training and testing sets. Therefore, using this method will give more credibility to the predicted results. Furthermore, all the datasets were normalized before modeling in the ranges $[-1, 1]$ in order to eliminate variations in input and output parameters. However, standardization of the dataset is not efficient, and the results are unfavorable due to an excessive distribution of the data.

Some statistical indices, such as coefficient of determination (R^2), root mean square error (RMSE) and mean absolute percentage error (MAPE), are used to determine the models' precision. The following formulas are used for calculating these indices (Eqs. 1-4).

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{sum of squares total (SST)}} \quad (1)$$

$$\text{MAE} = \left(\frac{1}{n}\right) \sum_{i=1}^n |y_i - y'_i| \quad (2)$$

$$\text{RMSE} = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y_i - y'_i)^2} \quad (3)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y'_i}{y_i} \right| \times 100\% \quad (4)$$

149 where y_i is the real value, y'_i is the forecasted value, $\bar{y}_i$ is the mean of y_i , $\bar{y}'_i$ is the mean of y'_i , and n is the number of
150 test datasets.

151 3. ML prediction models of slope stability and results analysis

152 Despite the merits of various ML approaches, according to the No-Free-Lunch (NFL) theorem, there is no ML model
153 to solve all engineering problems as the best method successfully. Therefore, researchers have tried to evaluate the
154 efficiency of various ML approaches for solving various optimization. (LSTM), (DNN), (KNN), (GPR), (SVR), and (DT) are
155 six separate ML and deep learning (DL) models with different features and capabilities that we use as NFL theorem.
156 However, the key features of the aforementioned models, which motivate us to use them, is as follows:

157 • Regression Analysis

158 Regression analysis is a statistical technique for predicting a continuous target variable from one or more
159 independent variables. Rather than variables that have been modified by experimentation, regression analysis is
160 typically used for naturally occurring variables. There are several different forms of regression, so how do we select
161 the one to use once we have chosen to use regression analysis?

162 We chose GPR because:

- 163 - GPR directly captures the model uncertainty. In regression, for example, GPR provides a distribution for the
164 prediction value rather than a single value as the prediction. In neural networks, this ambiguity is not explicitly
165 captured.
- 166 - By choosing different kernel functions, we can add prior knowledge and requirements about the shape of the
167 model while using GPR. For example, based on the answers to the following questions, we may choose
168 different priors. Is the model smooth? Is it sparse? Should it be able to change drastically? Should it be
169 differentiable? This capability gives researchers flexible models, which can be fitted to various kinds of
170 datasets.

171 • We chose SVR because:

172 The use of kernels, sparse solutions, and Vapnik-Chervonenkis (VC) regulation of the margin and number of support
173 vectors distinguish SVR. One of the key benefits of SVR is that its computational complexity is independent of the input
174 space's dimensionality. It performs lower computation compared to other regression techniques. It also has good
175 generalization, high prediction precision, and is resistant to outliers.

176 We chose DT because:

177 DTs are a form of supervision learning algorithm that partitions the sample multiple times based on specific
178 questions about the sample. For prediction problems, these can play a significant role. They are relatively simple to
179 comprehend and very operative. DTs reflect a series of decisions with varying chances of occurring. Through the use of
180 this technique, the identification of the most substantial variables and the relation between two or more variables can
181 be provided. In our problem, we have various variables, which are related to each other, so we select DT as one of
182 comparing models. In other words, a decision tree has the advantage of forcing the analysis of all possible outcomes
183 and tracing each direction to a conclusion. It performs a thorough analysis of the implications of each branch and
184 defines the required decision nodes.

185 Key advantages:

- 186 - There is no need to preprocess the data.
- 187 - Data distribution assumptions are not required.
- 188 - It can effectively handle collinearity.

189 - The prediction in a comprehensible manner can be obtained in DT.

190 • **We chose KNN because:**

191 In this paper, KNN were used as a non-parametric approach for prediction. It is one of the easiest ML approaches
192 that have been recently applied. It is a model of slow learning with local approximation. We use this model,
193 considering the following terms:

194 The key Advantages:

- 195 - It is a straightforward machine learning model and easy to use.
- 196 - It can tune only a few hyperparameters.

197 Hyperparameters:

- 198 - There are two main hyperparameters in KNN: K value & distance function.
- 199 - K value: the number of neighbors to include in the KNN algorithm. The value of k should be adjusted
200 according to the validation error.
- 201 - Distance function: The most commonly used similarity function is Euclidean distance. There are several
202 alternatives including the Manhattan distance, Hamming distance, and Minkowski distance.

203 Assumptions:

- 204 - A clear understanding of the input domain is required.
- 205 - A sample size should feasibly be moderate (due to space and time constraints).
- 206 - Prior to training, outliers and collinearity should be tackled.

207 Comparison with other models:

- 208 - The main difference between KNN and other models is that KNN needs a lot of real-time computation.

209 • **We chose DNN because:**

210 The substantial advantage of DNN over its predecessor is that it automatically identifies essential features without
211 the need for human intervention. In this case, we have no idea about the dependency of feature; therefore, we use
212 DNN as one of the comparators to decrease the dimensionality of datasets and see the effects of the automatic
213 dimensionality reduction in the results.

214 • **We chose LSTM because:**

215 The data of our problem has a time dependency feature. i.e., the variables are dependent on each other in terms of
216 time. Therefore, we decide to use RNN-based models to consider this time dependency. When we switch from RNN to
217 LSTM (Long Short-Term Memory), we introduce ever more control knobs that regulate the flow and mixing of inputs
218 according to qualified Weights. As a result, LSTM provides us with the most control and, consequently, better results.
219 However, this leads to an increase in complexity and operating costs. Consequently, in order to obtain the best
220 accuracy, we choose LSTM as a replacement for conventional RNN.

221 It is very significant to comprehend the data so that the right algorithm for the right problem can be chosen. In
222 some algorithms, only small sample sets are required, while others can work with tons and tons of samples. Some
223 certain algorithms desire numerical input while others work with categorial. Because our datasets are large, we
224 propose work with the LSTM Algorithm.

225 In the analysis of slope stability, GPR, SVR, and DT modeling are applied using MATLAB Software 2018. The LSTM,
226 DNN, and KNN algorithms are introduced in Anaconda version 3.6, and it a free, open-source application for scientific

computation in Python and R, which aims at simplifying package management and execution. These six predictive models have been introduced to determine the best predictive model for slope stability. In the following subsections, details on the ML prediction models and their prediction results are presented.

3.1 GPR

A Gaussian procedure (GP) is a gathering F of arbitrary factors $F_{x_1}, F_{x_2}, \dots$ for which any finite subset of the factors has a joint multivariate Gaussian conveyance. Components x of a set X is used to list the variables. For any finite length vector of lists $x = [x_1, x_2, \dots, x_n]^T$, we have a comparing vector $F_x = [F_{x_1}, F_{x_2}, \dots, F_{x_n}]^T$ of factors that has a multivariate Gaussian (or ordinary) distribution (Eq. 5) (Mahmoodzadeh et al., 2020a):

$$F_x \sim N\{\mu(x), k(x, x)\} \quad (5)$$

where the components of $\mu(x)$ are given by an earlier mean capacity $\mu(x_i)$, and k is the portion work. The portion uses two files x_i and x_j that provides the covariance between their comparing factors F_{x_i} and F_{x_j} . Given vectors of lists x_i and x_j , k returns the framework of covariances between all sets of factors where the first in the pair originates from F_{x_i} and the second from F_{x_j} . Each F_{x_i} is barely Gaussian, with a mean of $\mu(x_i)$ and difference of $k(x_i, x_i)$ (Mahmoodzadeh et al., 2020b).

Assume there has a capacity $f(x)$ that would want to upgrade. In addition to that, suppose that f could not be watched legitimately, yet that an arbitrary variable F_x can be seen that is listed by the same space as f and whose normal esteem is f , i.e., $\forall x \in X, E[F_x] = f(x)$. Particularly, it is accepted that the earlier conviction about the capacity f complies with a Gaussian procedure with earlier mean μ and part k . Furthermore, assume that F_x is a perception of $f(x)$ that has been tainted by zero-mean, i.i.d. Gaussian clamor, i.e., $F_x = f(x) + \epsilon$, where $\epsilon \sim N(0, \sigma_\epsilon^2)$. Consequently, $f(x)$ is a shrouded variable whose back appropriation and could be able to derive in the wake of watching tests of F_x at different areas in the space. The following subtraction is called Gaussian procedure relapse (Mahmoodzadeh et al., 2020c).

Give x a chance to be the arrangement of perceptions focuses and F_x be the subsequent genuine esteemed perceptions. This required to process the back appropriation of some new point $\hat{x} \in X$. The appropriation will be Gaussian with mean and difference (Eqs. 6 and 7).

$$\mu(\hat{x}|x) = \mu(\hat{x}) + k(\hat{x}, x)k(x, x)^{-1} (F_x - \mu(x)) \quad (6)$$

$$\sigma^2(\hat{x}|x) = k(\hat{x}, \hat{x}) - k(\hat{x}, x)k(x, x)^{-1} k(x, \hat{x}) \quad (7)$$

In the regression learner app embedded in MATLAB software 2018, four different models for GPR include squared exponential, rational quadratic, exponential, and Matern 5/2. After modeling by this program, the model type with the most accurate results is taken into account. Also, the optimization mode is considered in the app, so that the app itself optimizes the amount and type of hyper-parameters of the GPR model. The optimized type and value of the GPR hyper-parameters produced by the regression learner app are presented in Table 3.

The 5-fold CV results of slope stability predicted by the GPR model are depicted in Fig. 5. As in Fig. 5, the analyzed FOS values are very close to the FOS values predicted by the GPR model. According to Fig. 5, the FOS errors produced by the GPR model are close to zero for most datasets, so that the mean absolute error value is about 0.0842. Such an error in predicting the FOS of a slope is negligible and indicates the GPR model's high prediction accuracy. To more support these results, other statistical evaluation indices of R^2 (Fig. 6), RMSE, and MAPE are obtained equal to 0.8139,

0.160892561, and 7.209771731, respectively. As the database employed in this research, all the outcomes confirm the GPR model's high prediction accuracy.

3.2 SVR

Vapnik (1995) modified his first version model (ϵ -support vector regression, SVR) by changing the ϵ -insensitive loss function. This modification permits the SVR model to use the margin idea in the regression process. Margin in the modified model can be described by the summation of the distances of hyperplane from the closest points of two classes. Minimizing errors between the actual training data and the hyperplane are the main target of the SVR. Kernel function idea has introduced by Vapnik (1995) for non-linear SVR. Readers are directed to Vapnik (1995) to understand more about SVR.

To get the FOS predictions using the SVR model, in the MATLAB 2018 software, the regression learner app was embedded and applied. Six model types including cubic, linear, medium Gaussian, quadratic, fine Gaussian, and coarse Gaussian, are provided for the SVR method in MATLAB 2018. By optimizing the mode in the regression learner app, the type and values of the SVR hyper-parameters are obtained as in Table 4.

Fig. 7 shows the FOS predictions made by the SVR model. Compared to the analyzed FOS values in the PLAXIS software, there is acceptable accuracy in the predictions of the SVR model. As it turns out from Fig. 7, the FOS errors are often less than 0.2, so that the MAE value is equal to 0.142. This amount of error can be acceptable and significant in the FOS predictions of a slope. Compared to the analyzed FOS results, other statistical evaluation indices of R^2 , RMSE, and MAPE are obtained by 0.6950 (Fig. 8), 0.20650649, and 11.89200416%, respectively. All of these results suggest that the SVR model should be considered as a somewhat appropriate predictive model for predicting slope stability.

3.3 DT

The DT is one of the classifications and regression methods based on the non-parametric survived learning technique. Furthermore, it consists of a set of if-then-else decision rules. The best perdition of the model occurs when the DT goes deeper and deeper to make the best fit with the actual data. There are several advantages of the DT. First, the distribution of explanatory variables does not require assumption. Second, strong relations among independent variables do not affect the DT outcomes. Third, various dependent variables such as survived data, categorical and numerical can be covered by DT. Fourth, this technique comprises the powerful variables and eliminates the least powerful variables which describe the dependent variable. For the DT, it is possible to predict small and large datasets well, even though this technique was initially developed to only well predict large data (Mahmoodzadeh et al., 2020b).

The algorithm of DT can be explained as follow:

1. First, the calculation of the targeted variance is performed.
2. Based on the various attributes, the database is divided into distinct parts, and the variance of each sectioned part is deducted from the variance prior to the division. This can be defined as variance reduction.

Node N can be defined by the variance reduction as:

$$I_V(N) = \frac{1}{|S|^2} \sum_{i \in S} \sum_{j \in S} \frac{1}{2} (x_i - x_j)^2 - \left(\frac{1}{|S_t|^2} \sum_{i \in S_t} \sum_{j \in S_t} \frac{1}{2} (x_i - x_j)^2 + \frac{1}{|S_f|^2} \sum_{i \in S_f} \sum_{j \in S_f} \frac{1}{2} (x_i - x_j)^2 \right) \quad (8)$$

S is a group of samples that is not separated yet, S_t is a group of separated samples with true result and S_f is a group of separated samples with a false result. Without referring to the mean, each of the summands presented above is indeed variance estimates though written in a form. In each summation term in Equation 10, variance estimation is required in such a way the mean is not referred to directly.

The decided node of the attribute is based on the highest VR.

3. Depending on the values of selected attributes, the datasets are separated. If the variance of a part is more than zero, it is separated once more.
4. Keep another trial going until all the data is evaluated.

In the DT approach, three model, such as medium, coarse, and fine, are embedded in the MATLAB 2018. The FOS predictions were performed through these three models and eventually considered the model that provided more precise results. The information about the optimized DT's hyper-parameter parameters considered in this analysis is provided in Table 5.

In Fig. 9, the FOS results obtained from the DT model are compared with the analyzed ones. As shown in Fig. 9, there are small differences between the predicted outcomes and the analyzed values. The MAE is equal to 0.143. Achieving such an error in predicting the FOS of a slope is good. Other statistical evaluation indices of $R^2 = 0.6830$ (Fig.10), RMSE = 0.210590831, and MAPE = 12.2080868% also indicate the good accuracy in the DT model predictions.

3.4 LSTM

Deep learning can be modeled as a typical version of neural networks with several layers. The knowledge found within these deep learning networks is better remembered than traditional neural networks. A recurrent neural network (RNN) is a form of the combination of a loop network in the neural network. Knowledge persistence exists across these loops in the networks. This network gathers inputs and data from prior networks, implements specified procedures and provides the next network output. Some applications may only need current knowledge, while others may also need additional information from the past. A learning lag exists in redundant neural networks in which the gap between the desired input and the requirement is improved. However, LSTMs networks are a special kind of RNN (Mahmoodzadeh et al., 2021b) capable of studying these situations. Such networks are intentionally designed so that, on the permanent networks, the long-term dependency problem can be mitigated.

An essential function of LSTM is recalling knowledge over a long time. It is an ordinary choice since the precision of the model permanently depends on the sum of prior knowledge. The basic LSTM module, defined as the repetitive module, has a particular interaction with four layers with neural networks, as shown in Fig. 11. In the module, there are three gates with activation functions of σ_1 , σ_2 , and σ_3 , and it contains two output activation functions such as Φ_1 and Φ_2 , as seen in Fig. 11. To represent the addition and multiplication in the element-wise manner, the Σ and π symbols are used. A bullet ($\bullet$) symbol is used to describe concatenating mathematical procedures. The principal element of LSTM is the cell state, in which the current block memory (S_{t-1}) is created from the previous block memory. The information flow is then approved immediately down the line. In the network, the first layer (S_t) determines the sum of the prior knowledge to flow, and Eq. 9 demonstrates the activity carried out by this layer (Mahmoodzadeh et al., 2021c).

In the network, two layers and their associations influence the mechanism of storing new data in the cell state. A sigmoid layer (σ_2) that find out the (I_t) value to be updated see Eq. 10 and $\Phi_1 \tanh$ layer, which makes a new candidate value ($\tilde{S}_t$) as seen in Eq. 11. To make a case, both are to be added together with the state. The cell state is lastly replaced with Eq. 12.

$$c_{ft} = \sigma_1(W_{cf} \cdot [O_{t-1}, x_t] + b_{cf}) \quad (9)$$

$$I_t = \sigma_2(WI \cdot [O_{t-1}, x_t] + bI) \quad (10)$$

$$\tilde{S}_t = \tanh(WS \cdot [O_{t-1}, x_t] + bS) \quad (11)$$

$$S_t = (c_{ft} \times S_{t1}) + (I_t \times \tilde{S}_{t-1}) \quad (12)$$

In order to optimize the LSTM model, this model was tested with different types of parameters and changes in the number of neurons, layers, batch_size, epochs, and so on. Finally, the type/value of parameters of the LSTM were optimized as in Table 6.

The structure of the network optimized LSTM model is shown in Fig. 12.

The FOS results of LSTM are compared with the analyzed FOS results in Figs. 13 and 14. Given that deep learning methods such as LSTM require much more data to learn, it can be said that with the amount of data used in this article, still, it has provided acceptable accuracy. The MAE for the LSTM results is equal to 0.169. Achieving such an error in predicting the FOS of a slope is good. Other statistical evaluation indices of $R^2 = 0.6616$ (Fig.14), RMSE = 0.221912033 and MAPE = 14.52803317% also indicate good accuracy in the LSTM model predictions.

3.5 DNN

In the layered neural network of DNN, a deep feed-forward neural network is available. There is more than one layer of hidden units between its inputs and outputs. The layers contain input layers, which are followed by mid-layers, hidden layers, and finally, the output layer. Consequently, the input, hidden layers, and output layers are all connected to the network's neighboring layers. It is a feed-forward neural network, which means that there is no forward feeding and that the links between the layers are only one way, forward. Furthermore, DNN is especially well-suited to analyzing raw input data because it can recognize patterns and learn useful features from it without the need for rigorous feature engineering, data pre-processing, or hand-crafted guidelines. Moreover, with the rise in training data, its efficiency further improves. (Mahmoodzadeh et al., 2021b). DNNs have a much wide range of applications, from simple text creation to machine vision tasks, and the early uses of DNN are in automatic translation.

In order to optimize the DNN model, such as the LSTM model, it was tested with different types of parameters and changes in the number of neurons, layers, batch_size, epochs, and so on. Finally, the type/value of parameters of the DNN was optimized as in Table 7.

The network structure of the optimized LSTM model is illustrated in Fig. 15.

The predictions of FOS results of the DNN model are compared with the analyzed FOS results in Figs. 15 and 16. As the LSTM model, the DNN model also needs more data for training, but the amount of data used in this article has provided acceptable accuracy. The MAE for the DNN results is equal to 0.0.118. Achieving such an error in predicting the FOS of a slope is good and acceptable. Other statistical evaluation indices of $R^2 = 0.7498$ (Fig.16), RMSE = 0.188416005, and MAPE = 9.967075268% also indicate the good accuracy in the DNN model predictions.

3.6 KNN

KNN is a simple and well-managed machine learning algorithm that can be applied to classification and regression problems. It is important to apply and comprehend, but it is very difficult to do as the amount of data in use expands considerably. KNN functions through the determination of the ranges between a query and all the examples in the results, choosing the specified number of examples for the query (K), then voting for the most frequent label (in the situation of classifying) or averaging the labels (In the situation of a regression). During the regression and classification process, selecting the correct K for the data is achieved by trying multiple Ks and finding the one that fits

377 better (Mahmoodzadeh et al., 2020c). To implement the KNN method in this study, Anaconda version 3.6 is used.
378 During the KNN model's predictive analysis, various Ks were evaluated, and the best predictions were obtained with K
379 = 2.

380 For the KNN model, as shown in Fig. 18, the difference in the predicted FOS values with the analyzed results is low.
381 This MAE = 0.0913 indicates the high accuracy of the predictions made by the KNN model. The results of other
382 statistical evaluation indices of $R^2 = 0.7080$ (Fig. 19), RMSE = 0.205355281, and MAPE = 7.874855748% also indicate
383 the high precision of the KNN model in predicting the slope stability. Thus, considering the database employed in this
384 paper, the KNN model is a significant and considerable method for slope stability prediction.

385 To determine the best prediction model among the six ML models used in this paper to predict the slope stability,
386 in Fig. 20 and Table 6, a comparison between the results predicted by them has been made. By analyzing and
387 comparing the values of the obtained statistical evaluation indices for each model, it can be concluded that almost all
388 models have provided close results to each other, and there is not much difference between their accuracy. However,
389 the highest accuracy is produced by the GPR model. Therefore, the most acceptable results, which are not very
390 different from the analyzed ones, are provided by the GPR model.

391 4. Discussion

392 In this study, the generalization of the suggested Gaussian process regression methodology is discussed.
393 Generalization is a concept used to characterize the model's ability to interact and adapt to new information.
394 Therefore, a model can ingest novel data and predict accurately after practicing with data not used during training.
395 The basis for a model's success and its practical performance is related to its capability to generalize. When a model
396 was so well trained in training data, it cannot be generalized. When new data are given, the model is rendered
397 inaccurate predictions and worthless even if it can accurately predict the training data. A model starts 'memorizing'
398 the training data instead of 'learning'; this is known as overfitting.

399 Feature selection can be used to avoid the overfitting of the model. In this case, feature selection would minimize the
400 number of features, which decreases the computational complexity of the model. The stepwise approach for choosing
401 an important collection of features from the data sets is used for the total features available.

402 Stepwise regression methods can be classified into three strategies:

- 403 • **The first strategy (forward selection)**, which begins without any predictors in the model, relies on adding
404 more iterative predictors to the model. Simultaneously, it stops when the improvement in the results no
405 longer has a statistically positive impact
- 406 • **The second strategy (backward selection)**, which begins with all predictors in the model, periodically
407 eliminates the lowest contributive predictors; while it stops once you get a model, all its predictors become
408 statistically meaningful.
- 409 • **The third strategy (stepwise selection)** is a mixture of forwarding and backward processes. It begins without
410 any predictors and then adds the predictors that contribute most to the outcome sequentially (like backward
411 selection). While adding every new variable, those that no more enhance the model's fit should be removed
412 (like forwarding selection).

413 In this research, the step AIC [MASS Package] was used, which defines the best design by AIC. The model also has a
414 choice called direction, which takes these values: i) forward (for elimination from forwarding); ii) backward (for
415 elimination from backward); iii) both (sequential replacement, for forward and backward elimination). The best-
416 finished model is recovered. In R, among the most popular search methods for selecting features is stepAIC. For the
417 stepAIC model's values continuously to arrive at the final feature set are attempted to be reduced. In the tables below,
418 the finding listed as follows, three asterisks (*) reflect the highly significant value of p. Therefore, it could reject the

419 null hypothesis by providing a small p-value for the intercept and path that enables us to create a good relationship
420 between two measured variables (the target and the predictor variables). A p-value around 5% or less would be a
421 good cut-off point for most situations.

422 In the first step, we fit the model with all predictors and targets. The first step results of feature selection are
423 presented in Table 7. As in Table 7, the effect of all features on slope stability is significant. In this case, we do not
424 delete any of the features, and we stop at this step. According to Table 7, the most effective parameters on slope
425 stability with respect to the p-values are ϕ and c . Also, the parameter γ has the least impact on slope stability.

426 5. Conclusions

427 This study proposed six ML models to predict slope stability. 327 datasets analyzed using PLAXIS software including
428 six input parameters of γ , c , ϕ , α , H , and r_u , and one target of FOS were employed in the models. The 5-fold CV method
429 was used to evaluate the prediction performance of the models.

430 All the models produced acceptable results and almost close to each other. However, the GPR model with $R^2 =$
431 0.8139 , $RMSE = 0.160893$, and $MAPE = 7.209772\%$, was the most accurate model to predict slope stability.

432 The backward selection method was used to assess the contribution of each parameter in the prediction problem.
433 The results showed that all the features have significant contributions to slope stability. However, features ϕ and γ
434 were the most effective and least effective parameters on slope stability, respectively.

435 Conflict of interest

436 There is no conflict of interest.

437 References

- 438 Sakellariou MG, Ferentinou MD. A study of slope stability prediction using neural networks. *Geotech. Geol. Eng.* 23
439 (2005) 419-445.
- 440 Duncan JM. Factors of safety and reliability in geotechnical engineering. *J. Geotech. Geoenviron. Eng.* 126 (2000) 307-
441 316.
- 442 Suman S, Khan SZ, Das SK, Chand SK. Slope stability analysis using artificial intelligence techniques. *Natural Hazards* 84
443 (2016) 727-748.
- 444 Bye AR, Bell FG. Stability assessment and slope design at Sandsloot open pit, South Africa. *Int. J. Rock Mech. Mining*
445 *Sci.* 38 (2001) 449-466.
- 446 Trivedi R, Vishal V, Pradhan S, Singh T, Jhanwar J. Slope stability analysis in limestone mines. *Int. J. Earth Sci. Eng.* 5
447 (2012) 759-766.
- 448 Thiebes B, Bell R, Glade T, Jäger S, Anderson M, Holcombe L. A WebGIS decision-support system for slope stability
449 based on limit-equilibrium modelling. *Eng. Geol.* 158 (2013) 109-118.
- 450 Thiebes, B., Bell, R., Glade, T. et al. Integration of a limit-equilibrium model into a landslide early warning system.
451 *Landslides* 11, 859–875 (2014). <https://doi.org/10.1007/s10346-013-0416-2>
- 452 He, L., Wu, G. & Wang, H. Study of base friction simulation tests based on a complicated engineered bridge slope.
453 *Front. Struct. Civ. Eng.* 6, 393–397 (2012). <https://doi.org/10.1007/s11709-012-0174-6>
- 454 Kim JC, Jung H, Kim S, Chung K. Slope based intelligent 3D disaster simulation using physics engine. *Wireless Pers.*
455 *Commun.* 86 (2016) 183-199.

456 Sarkar, K., Singh, T.N. & Verma, A.K. A numerical simulation of landslide-prone slope in Himalayan region—a case
457 study. *Arab J Geosci* 5, 73–81 (2012). <https://doi.org/10.1007/s12517-010-0148-8>

458 Pirone, M., Papa, R., Nicotera, M.V. et al. In situ monitoring of the groundwater field in an unsaturated pyroclastic
459 slope for slope stability evaluation. *Landslides* 12, 259–276 (2015). <https://doi.org/10.1007/s10346-014-0483-z>

460 Liu Z, Shao J, Xu W, Shi C. Estimation of Elasticity of Porous Rock Based on Mineral Composition and Microstructure.
461 *Advances in Materials Science and Engineering*, vol. 2013, Article ID 512727, 10 pages, 2013.
462 <https://doi.org/10.1155/2013/512727>

463 Liu Z, Shao JF, Weiya X, Wu Q. Indirect estimation of unconfined compressive strength of carbonate rocks using
464 extreme learning machine. *Acta Geotechnica*, Springer Verlag, 2015, 10 (5), pp.651-663. [https://hal.archives-](https://hal.archives-ouvertes.fr/hal-01200765)
465 [ouvertes.fr/hal-01200765](https://hal.archives-ouvertes.fr/hal-01200765)

466 Youssef, A.M., Pourghasemi, H.R., Pourtaghi, Z.S. et al. Landslide susceptibility mapping using random forest, boosted
467 regression tree, classification and regression tree, and general linear models and comparison of their performance
468 at Wadi Tayyah Basin, Asir Region, Saudi Arabia. *Landslides* 13, 839–856 (2016). [https://doi.org/10.1007/s10346-](https://doi.org/10.1007/s10346-015-0614-1)
469 [015-0614-1](https://doi.org/10.1007/s10346-015-0614-1)

470 Zhou J, Li X, Mitri HS. Classification of Rockburst in Underground Projects: Comparison of Ten Supervised Learning
471 Methods. *Journal of Computing in Civil Engineering* 30 (2016)
472 <https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29CP.1943-5487.0000553>

473 Lu, P. and Rosenbaum, M. S. (2003). Artificial neural networks and grey systems for the prediction of slope stability.
474 *Natural Hazards*, Vol. 30, No. 3, pp. 383-398, DOI: 10.1023/B:NHAZ.0000007168.00673.27.

475 Zhao, H., Yin, S., Ru, Z., 2012. Relevance vector machine applied to slope stability analysis. *Int. J. Numer. Anal.*
476 *Methods Geomech.* 36 (5), 643–652.

477 Li S, Zhao HB, Ru Z. Slope reliability analysis by updated support vector machine and Monte Carlo simulation. *Natural*
478 *Hazards* 65 (2013) 707-722.

479 Wen T, Zhang B, Prediction model for open-pit coal mine slope stability based on random forest. *Sci. Technol. Rev.* 32
480 (2014) 105-109.

481 Khajehzadeh M, Taha MR, El-Shafie A, Eslami M. A modified gravitational search algorithm for slope stability analysis.
482 *Eng. Appl. Artif. Intell.* 25 (2012) 1589-1597, Dec. 2012.

483 Raihan TM, Mohammad K, Mahdijeh E. A new hybrid algorithm for global optimization and slope stability evaluation.
484 *J. Central South Univ.* 20 (2013) 3265-3273.

485 Shi XZ, Zhou J, Zheng W, Hu H, Wang HY. Bayes discriminant analysis method and its application for prediction of slope
486 stability. *J. Sichuan Univ., Eng. Sci. Ed.* 42 (2010) 63-68.

487 Feng, X. (2000). *Introduction of Intelligent Rock Mechanics*, Science Press, Beijing, China, pp. 239-241.

488 Li, X. (2004). Comparative studies of artificial neural networks and adaptive Neuro-Fuzzy inference system based
489 approach for the circular sliding slopes stability analysis, Master Thesis, University of South China, Hengyang,
490 Hunan, China.

491 Huang, Z., Cui, J., and Liu, H. (2004). Chaotic neural network method for slope stability prediction. *Chinese Journal of*
492 *Rock Mechanics and Engineering*, Vol. 22, pp. 015.

493 Wang, H., Xu, W., and Xu, R. (2005). Slope stability evaluation using back propagation neural networks. *Engineering*
494 *Geology*, Vol. 80, No. 3, pp. 302-315, DOI: 10.1016/j.enggeo.2005.06.005.

495 Samui, P. (2008). Slope stability analysis: A support vector machine approach. *Environmental Geology*, Vol. 56, No. 2,
496 pp. 255-267, DOI: 10.1007/s00254-007-1161-4.

497 Zhao, H.-b. (2008). Slope reliability analysis using a support vector machine. *Computers and Geotechnics*, Vol. 35, No.
498 3, pp. 459-467, DOI: 10.1016/j.compgeo.2007.08.002.

499 Choobbasti, A., Farrokhzad, F., and Barari, A. (2009). Prediction of slope stability using artificial neural network (case
500 study: Noabad, Mazandaran, Iran). *Arabian Journal of Geosciences*, Vol. 2, No. 4, pp. 311-319, DOI:
501 10.1007/S12517-009-0035-3.

502 Das, S. K., Biswal, R. K., Sivakugan, N., and Das, B. (2011). "Classification of slopes and prediction of factor of safety
503 using differential evolution neural networks." *Environmental Earth Sciences*, Vol. 64, No. 1, pp. 201-210, DOI:
504 10.1007/S12665-010-0839-1.

505 Erzin, Y. and Cetin, T. (2013). The prediction of the critical factor of safety of homogeneous finite slopes using neural
506 networks and multiple regressions. *Computers & Geosciences*, Vol. 51, pp. 305-313, DOI:
507 10.1016/j.cageo.2012.09.003.

508 Liu, Z., Shao, J., Xu, W., Chen, H., and Zhang, Y. (2014). "An extreme learning machine approach for slope stability
509 evaluation and prediction." *Natural Hazards*, Vol. 73, No. 2, pp. 787-804, DOI: 10.1007/s11069-014-1106-7.

510 Gordan, B., Jahed Armaghani, D., Hajihassani, M., and Monjezi, M. (2016). Prediction of seismic slope stability through
511 combination of particle swarm optimization and neural network. *Engineering with Computers*, Vol. 32, No. 1, pp.
512 85-97, DOI: 10.1007/s00366-015-0400-7.

513 Hoang, N.-D. and Pham, A.-D. (2016). Hybrid artificial intelligence approach based on metaheuristic and machine
514 learning for slope stability assessment: A multinational data analysis. *Expert Systems with Applications*, Vol. 46, pp.
515 60-68, DOI: 10.1016/j.eswa.2015.10.020.

516 Suman, S., Khan, S., Das, S., and Chand, S. (2016). Slope stability analysis using artificial intelligence techniques.
517 *Natural Hazards*, Vol. 84, No. 2, pp. 727-748, DOI: 10.1007/s11069-016-2454-2.

518 Verma, A., Singh, T., Chauhan, N. K., and Sarkar, K. (2016). A hybrid FEM-ANN approach for slope instability prediction.
519 *Journal of The Institution of Engineers (India): Series A*, Vol. 97, No. 3, pp. 171-180, DOI: 10.1007/s40030-016-0168-
520 9.

521 Rukhaiyar, S., Alam, M., and Samadhiya, N. (2017). A PSO-ANN hybrid model for predicting factor of safety of slope.
522 *International Journal of Geotechnical Engineering*, pp. 1-11, DOI: 10.1080/19386362.2017.1305652.

523 Xue, X. (2017). Prediction of slope stability based on Hybrid PSO and LSSVM. *Journal of Computing in Civil Engineering*,
524 Vol. 31, No. 1, pp. 1-10, DOI: 10.1061/(ASCE)CP.1943-5487.0000607.

525 Feng X, Li S, Yuan C, Zeng P, Sun Y. Prediction of Slope Stability using Naive Bayes Classifier. *KSCE Journal of Civil*
526 *Engineering* 22 (2018): 941-950.

527 Zhou J, Li E, Yang S, Wang M, Shi X, Yao S, Mitri HS, Slope stability prediction for circular mode failure using gradient
528 boosting machine approach based on an updated database of case histories, *Safety Science* 118 (2019) 505-518.
529 <https://doi.org/10.1016/j.ssci.2019.05.046>.

530 Zhang Z, Liu Z, Zheng L, Zhang Y. Development of an adaptive relevance vector machine approach for slope stability
531 inference, *Neural Comput. Appl.*, vol. 25, pp. 2025-2035, Dec. 2014.

532 Lin Y, Zhou K, Li J. Prediction of Slope Stability Using Four Supervised Learning Methods. In *IEEE Access*, vol. 6, pp.
533 31169-31179, 2018, doi: 10.1109/ACCESS.2018.2843787.

Manouchehrian A, Gholamnejad J, Sharifzadeh M. Development of a model for analysis of slope stability for circular mode failure using genetic algorithm," *Environ. Earth Sci.*, vol. 71, pp. 1267-1277, Feb. 2014.

Xue, X., Yang, X. & Chen, X. Application of a support vector machine for prediction of slope stability. *Sci. China Technol. Sci.* **57**, 2379–2386 (2014). <https://doi.org/10.1007/s11431-014-5699-6>

Mahmoodzadeh A, Zare S (2016) Probabilistic prediction of expected Ground conditions and construction time and costs in road tunnels. *Journal of Rock Mechanics and Geotechnical Engineering* 8:734-745. doi: 10.1016/j.jrmge.2016.07.0 01

Mahmoodzadeh A, Mohammadi M, Daraei A, Rashid T.A, Sherwani A.F.H, Faraj R.H, Darwesh A.M. Updating ground conditions and time-cost scatter-gram in tunnels during excavation, *Autom. Constr.* 105 (2019) 102822. <https://doi.org/10.1016/j.autcon.2019.04.017>.

Mahmoodzadeh A, Mohammadi M, Daraei A, Faraj R.H, Omer R.M.D, Sherwani A.F.H. Decision-making in tunneling using artificial intelligence tools. *Tunnelling and Underground Space Technology* (2020a). <https://doi.org/10.1016/j.tust.2020.103514>

Mahmoodzadeh A, Mohammadi M, Daraei A, Hama-Ali H.F, Abdullah A.I, Al-Salihi N.K. Forecasting tunnel geology, construction time and costs using machine learning methods. *Neural Computing and Applications* (2020b). DOI: 10.1007/s00521-020-05006-2

Mahmoodzadeh A, Mohammadi M, Daraei A, Hama-Ali H.F, Al-Salihi N.K, Omer R.M.D (2020c) Forecasting maximum surface settlement caused by urban tunneling. *Automation in Construction* 120:103375. <https://doi.org/10.1016/j.autcon.2020.103375>

Mahmoodzadeh A, Mohammadi M, Abdulhamid SN, Ibrahim HH, Hama Ali HF, Salim SG. Dynamic reduction of time and cost uncertainties in tunneling projects. *Tunnelling and Underground Space Technology* 109 (2021a) 103774. <https://doi.org/10.1016/j.tust.2020.103774>.

Mahmoodzadeh A, Mohammadi M, Ibrahim HH, Abdulhamid SN, Salim SG, Hama Ali HF, Majeed MK. Artificial intelligence forecasting models of uniaxial compressive strength. *Transportation Geotechnics* 27 (2021b) 100499. <https://doi.org/10.1016/j.trgeo.2020.100499>.

Mahmoodzadeh A, Mohammadi M, Hama Ali HF, Abdulhamid SN, Ibrahim HH, Noori KMG. Dynamic prediction models of rock quality designation in tunneling projects. *Transportation Geotechnics* 27 (2021c) 100497. <https://doi.org/10.1016/j.trgeo.2020.100497>.

Mahmoodzadeh A, Mohammadi M, Ibrahim HH, Noori KMG, Abdulhamid SN, Hama Ali HF. Forecasting sidewall displacement of underground caverns using machine learning techniques. *Automation in Construction* 123 (2021d) 103530. <https://doi.org/10.1016/j.autcon.2020.103530>.

Vapnik, V.N., 1995. The nature of statistical learning theory. New York: Springer.

Figure captions

- Fig. 1** Overall procedure of slope stability prediction using ML techniques.
- Fig. 2** A slope geometry and the effective parameters on the stability of slopes.
- Fig. 3** The box graph of six factors.
- Fig. 4** Pie chart of all slope cases.
- Fig. 5** Comparison of the FOS results predicted by the GPR model with the analyzed ones.
- Fig. 6** FOS results produced by the GPR model versus the analyzed results.
- Fig. 7** Comparison of the SVR results with the analyzed ones.

574 **Fig. 8** FOS results produced by the SVR model versus the analyzed results.
575 **Fig. 9** Comparison the FOS results predicted by the DT model with the analyzed ones.
576 **Fig. 10** FOS results produced by the DT model versus the analyzed results.
577 **Fig.11** LSTM repeating module (Mahmoodzadeh et al., 2021b).
578 **Fig 12.** The LSTM model network used in this study.
579 **Fig. 13** Comparison the FOS results predicted by the LSTM model with the analyzed ones.
580 **Fig. 14** FOS results produced by the LSTM model versus the analyzed results.
581 **Fig 15.** The DNN model network used in this study.
582 **Fig. 16** Comparison of the FOS results predicted by the DNN model with the analyzed ones.
583 **Fig. 17** FOS results produced by the DNN model versus the analyzed results.
584 **Fig. 18** Comparison of the FOS results predicted by the KNN model with the analyzed ones.
585 **Fig. 19** FOS results produced by the KNN model versus the analyzed results.
586 **Fig. 20** Comparison between the R^2 of the prediction models.

587

588 **Table captions**

589 **Table 1** Earlier studies on the slope stability predictions using soft computing approaches.
590 **Table 2** Statistical features of the remained slope cases.
591 **Table 3** The optimized parameters of the GPR model.
592 **Table 4** The optimized parameters of the SVR model.
593 **Table 5** The optimized model type and hyper-parameters of the DT method.
594 **Table 6** The type/value of the optimized LSTM parameters.
595 **Table 7** The type/value of the optimized DNN parameters.
596 **Table 8** Comparison of the statistical evaluation indices results produced by the prediction models.
597 **Table 9** The first step of feature selection.

598

Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

Figure 6

Figure 7

Figure 8

Figure 9

Figure 10

Figure 11

Figure 12

Figure 13

Figure 14

Figure 15

Figure 16

Figure 17

Figure 18

Figure 19

Figure 20

Table 1

References	Parameters	Soft computing method	Datasets no.
Feng, (2000)	$r_u, H, \alpha, \gamma, \phi, c$	FFNN	82
Lu and Rosenbaum, (2003)	$H, \alpha, \gamma, \phi, c, r_u$	BPNN	32
Li, (2004)	$H, \alpha, \gamma, \phi, c$	ANN and ANFIS	59
Huang et al., (2004)	$H, \alpha, \gamma, \phi, c, r_u$	CNN	64
Sakellariou and Ferentinou, (2005)	$H, \alpha, \gamma, \phi, c, r_u$	BPNN	46
Wang et al., (2005)	$H, \alpha, \gamma, \phi, c$	BPNN	27
Samui, (2008)	$H, \alpha, \gamma, \phi, c, r_u$	SVM	46
Zhao, (2008)	γ, ϕ, c	SVM	10
Choobbasti et al., (2009)	$a_h, X, \phi, \alpha, a_v, c, Y, S_r$	ANN	36
Das et al., (2011)	$r_u, H, \alpha, \gamma, \phi, c$	ANN	46
Erzin and Cetin, (2013)	$\gamma, c, \phi, \alpha, H$	ANN	675
Liu et al., (2014)	$r_u, H, \alpha, \gamma, \phi, c$	ELM	97
Gordan et al., (2016)	$H, \alpha, \gamma, \phi, c, PGA$	PSO-ANN	699
Hoang and Pham, (2016)	$r_u, H, \alpha, \gamma, \phi, c$	LS-SVC	168
Suman et al., (2016)	$r_u, H, \alpha, \gamma, \phi, c$	FNs, MARS, MGGP	103
Verma et al., (2016)	pp, c, α, ϕ	ANN	100
Rukhaiyar et al., (2017)	$r_u, H, \alpha, \gamma, \phi, c$	PSO-ANN	83

Xue, (2017)	$r_u, H, \alpha, \gamma, \phi, c$	PSO–LSSVM	46
Feng et al., (2018)	$r_u, H, \alpha, \gamma, \phi, c$	NBC	82
Zhou et al., (2019)	$r_u, H, \alpha, \gamma, \phi, c$	GBM	221
Ray et al., (2020)	$H, c_{r_i}, \phi_{j_i}, c_{s_i}, \phi_{s_i}, c_{j_i}, D, E_{r_i}, \phi_{r_i}, \alpha$	ANN	400

Table 2

	γ [kg/m ³]	c [kPa]	Φ [°]	α [°]	H [m]	r_u	FOS
count	327.0000	327.0000	327.0000	327.0000	327.0000	327.0000	327.0000
mean	20.50721	16.15281	27.75079	34.89785	78.90743	0.192095	1.258752
std	4.092716	15.88120	10.40078	10.35574	114.9846	0.175172	0.372882
min	12.00000	0.000000	0.000000	16.00000	3.600000	0.000000	0.236000
25%	18.50000	3.400000	22.00000	28.00000	10.00000	0.000000	1.020000
50%	20.00000	11.71000	30.00000	35.00000	32.00000	0.250000	1.200000
75%	22.40000	24.90000	35.00000	45.00000	88.00000	0.320000	1.450000
max	31.30000	70.00000	45.00000	65.00000	565.0000	0.640000	2.310000

Table 3

Parameter	Type/Value
Kernel function	'Exponential'
Beta	1.2517
Fit method	Exact Gaussian process regression
Sigma	0.0845
Basis function	'Constant'

Table 4

Parameter	Value or type
Kernel function	'Medium Gaussian'
Solver	'SMO'
Epsilon	0.0319
Bias	1.3317

Table 5

Parameter	Type/Value
Prune	on
Model type	Fine tree
Maximum number of splits	326
Predictor selection	allsplits
Minimum leaf size	4
Split criterion	mse
Minimum ParentSize	10

Table 6

No. of layers	Model type	Neurons no.	Activation function	Optimizer	batch_size	Epochs
1 input layer	Sequential()	6 neurons for the input layer	ReLU	Nadam	10	100
4 hidden layers		32 neurons within each hidden layer				
1 output layer		1 neuron for the output layer				

663

664

665 **Table 7**

No. of layers	Model type	Neurons no.	Activation function	Optimizer	batch_size	Epochs	Kernel initializer
1 input layer 4 hidden layers 1 output layer	Sequential()	Input layer: 6 neurons Hidden layers: 32 neurons Output layer: 1 neuron	ReLU	Nadam	10	100	Lecun_uniform

666

667

668 **Table 8**

Method	R ²	MAE	RMSE	MAPE [%]
GPR	0.8139	8.42E-02	0.160892561	7.209771731
SVR	0.6950	1.42E-01	0.20650649	11.89200416
DT	0.6830	1.43E-01	0.210590831	12.2080868
LSTM	0.6616	1.69E-01	0.221912033	14.52803317
DNN	0.7498	1.18E-01	0.188416005	9.967075268
KNN	0.7080	9.13E-02	0.205355281	7.874855748

669

670

671 **Table 9**

	Estimate	SE	tStat	p-value	Significance code
(Intercept)	1.3161	0.074702	17.619	4.85E-49	***
γ	0.011828	0.0036237	3.2642	0.0012166	**
c	0.023571	0.0014477	16.281	7.75E-44	***
ϕ	-0.025905	0.0014063	-18.421	3.63E-52	***
α	-0.00088078	0.00012506	-7.0428	1.16E-11	***
H	0.0094102	0.00084569	11.127	1.57E-24	***
r_u	-0.72688	0.066114	-10.994	4.57E-24	***

672

673