

Article

Machine-Learning-Based Classification Approaches toward Recognizing Slope Stability Failure

Hossein Moayed ^{1,2} , Dieu Tien Bui ^{3,4,*} , Bahareh Kalantar ^{5,*}  and Loke Kok Foong ⁶

¹ Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Vietnam; hossein.moayed@tdtu.edu.vn

² Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

³ Institute of Research and Development, Duy Tan University, Da Nang, Vietnam

⁴ Geographic Information System Group, Department of Business and IT, University of South-Eastern Norway, N-3800 Bø i Telemark, Norway

⁵ RIKEN Center for Advanced Intelligence Project, Goal-Oriented Technology Research Group, Disaster Resilience Science Team, Tokyo 103-0027, Japan

⁶ Centre of Tropical Geoenvironment (Geotropik), School of Civil Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, 81310 Johor, Malaysia; kfloke2@live.utm.my

* Correspondence: buitiendieu@duytan.edu.vn (D.T.B.); bahareh.kalantar@riken.jp (B.K.)

Received: 7 October 2019; Accepted: 26 October 2019; Published: 31 October 2019



Abstract: In this paper, the authors investigated the applicability of combining machine-learning-based models toward slope stability assessment. To do this, several well-known machine-learning-based methods, namely multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT), were selected to evaluate the stability of a slope through estimating the factor of safety (FOS). In the following, a comparative classification was carried out based on the five stability categories. Based on the respective values of total scores (the summation of scores obtained for the training and testing stages) of 15, 35, 48, 15, 50, 60, and 57, acquired for MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT, respectively, it was concluded that RF outperformed other intelligent models. The results of statistical indexes also prove the excellent prediction from the optimized structure of the ANN and RF techniques.

Keywords: slope stability; machine learning; classification; failure recognition

1. Introduction

The stability of local slopes is a critical issue that needs to be meticulously investigated due to their great impact on the adjacent engineering projects (excavation and transmission roads, for instance). In addition to that, slope failures cause great psychological damage (e.g., the loss of property and human life) all over the world every year. In Iran, for example, based on a rough estimation announced by the Iranian Landslide Working Party (2007), 187 people have been killed every year by slope failure in this country [1]. The saturation degree, as well as other intrinsic properties of the soil, significantly affect the likelihood of slope failure [2,3]. Up to now, many scientists have intended to provide effective modeling for slope stability problems. Some drawbacks of traditional methods, such as the necessity of using laboratory equipment and the high level of complexity prevent them from being a simply used solution [4]. Moreover, they cannot be used as a comprehensive solution due to their limitation in investigating a specific slope condition (e.g., slope angle, height, groundwater level, soil properties, etc.). Different types of limit equilibrium methods (LEMs), finite element model (FEM), and numerical solutions have been extensively employed for the engineering problem [5–8].

Development of design charts has been investigated for years in order to provide a reliable tool for slope stability assessment [9]. However, this approach is not devoid of defects either. Generating an efficient design chart requires consuming a lot of time and cost. Furthermore, determining the precise mechanical parameters is a difficult task [10,11]. Hence, the application of artificial intelligence techniques is more highlighted due to their quick performance and sufficient accuracy. The prominent superiority of these approaches is that they are able to detect the non-linear relationship between the target parameter(s) and its key factors. Also, ANN can be implemented using any determined number of hidden nodes [12,13]. For example, the suitable efficiency of an artificial neural network (ANN) [14–16] and a support vector machine (SVM) [17] has been proven in many geotechnical studies. After reviewing the literature, the complexity of the slope stability problem is evident. What makes the issue even more complicated and critical is constructing various structures in the vicinity of slopes (i.e., indicating a noticeable number of loads applied on a rigid footing). The amount of surcharge and the distance from the slope's crest are two parameters influencing the stability of the purposed slope [18]. This encouraged scholars to present an equation to calculate the factor of safety (FOS) of pure slopes or the slopes receiving a static load [19–23]. Prior to this study, Chakraborty and Goswami estimated the FOS for 200 slopes with different geometric and shear strength parameters using the multiple linear regression (MLR) and ANN models. They also compared the obtained results with an FEM model, and an acceptable rate of accuracy was obtained for both applied models. In this research, ANN outperformed MLR. Also, Pei et al. [23] effectively used the random forest (RF) and regression tree predictive models in functional soil-landscape modeling for regionalizing the depth of the failure plane and soil bulk density. They developed a classification for detecting the safe and hazardous slopes by means of FOS calculations. In some cases, the random forest and logistic regression techniques were implemented for a susceptibility modeling of a shallow landslide phenomena. They found that applied models have an excellent capability for this work. The SVM model synthesized by a minimal sequential optimization (SMO) algorithm has been used for surface grinding quality [24] and many medical classifications [25]. Also, a radial basis function regression (RBFR) has been successfully employed in genetics for predicting quantitative traits [26], and in agriculture for the discrimination of cover crops in olive orchards [27]. Another machine learning method that is used in the current paper is lazy k-nearest neighbor (IBK) tool. As the name implies, the lazy learning is the essence of this model and has shown good performance for many fields, such as spam e-mail filtering [28] and land cover assessment [29].

Though plenty of effort has been carried out to employ various soft computing tools for evaluating the stability of slopes [30,31], no prior study was found to address a comprehensive comparison of the applicability of the mentioned models within the same paper. Therefore, in this paper, we investigated the efficiency of seven machine-learning models (i.e., in their optimal structure), namely multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT), for appraising the stability of a cohesive soil slope. Furthermore, since the authors did not find any former application of IBK, RBFR, or SMO-SVM for estimating the FOS of the slope, the further novelty of the current study can be announced as applying the mentioned models for the first time in this field. In this regard, the Optum G2 software was used to give the FOS of 630 different slopes. Four conditioning factors that affected the values of FOS were chosen: undrained shear strength (C_u), slope angle (β), setback distance ratio (b/B), and applied surcharge on the shallow foundation installed over the slope (w). The utilized models were implemented in Waikato Environment for Knowledge Analysis (WEKA) software (stable version, Waikato, New Zealand) using the training dataset (i.e., 80% of the entire dataset). Then, the performance of each model was evaluated using testing samples (i.e., 20% of the entire dataset). The results were presented by means of several statistical indices. In the end, a new classification was carried out based on different degrees of stability to evaluate and compare the results of the models used. In addition, the design solution charts were developed using the outputs of the most efficient model.

2. Data Collection and Methodology

To acquire a reliable dataset, the use of a single-layer slope was proposed. It was supposed that a purely cohesive soil, which only had undrained cohesive strength (C_u), comprises the body of this slope (see Figure 1). The main factors that affect the strength of the slope against the failure (i.e., FOS) were considered to be the slope angle (β), setback distance ratio (b/B), and applied surcharge on the footing installed over the slope (w). These parameters are illustrated in Figure 1. To calculate the FOS, Optum G2 software, which is a comprehensive finite element program for geotechnical stability and deformation analysis, was used [32]. Regarding the possible values for the strength of the soil (C_u) and applied surcharge (w), different geometries of the slope (β), and displacements of the rigid foundation (b/B), 630 possible stages were modelled and analyzed in Optum G2 to calculate the FOS of each stage as the output. In the modeling process, the mechanical parameters of Poisson's ratio, soil unit weight, and internal friction angle were assigned to be 0.35, 18 kN/m³, and 0°, respectively. Also, Young's modulus (E) was varied for every value of C_u . In this sense, E was set to 1000, 2000, 3500, 5000, 9000, 15,000, and 30,000 kPa for the following respective values of C_u : 25, 50, 75, 100, 200, 300, and 400 kPa.

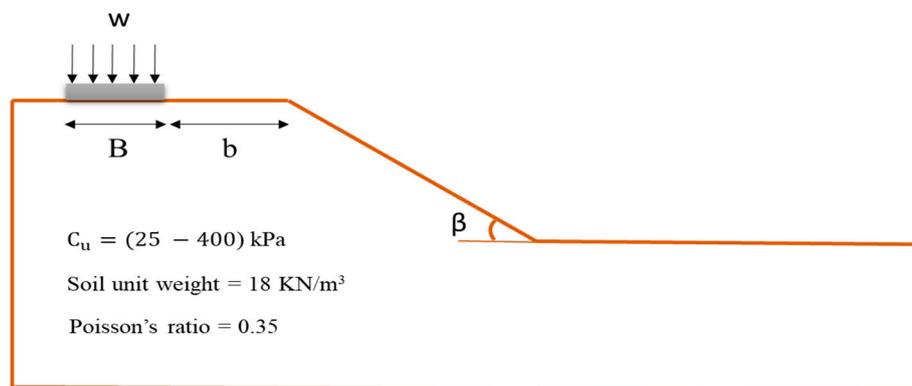


Figure 1. A view of the geometry of the slope.

The graphical relationship between the FOS and its influential factors is depicted in Figure 2a–d. In these charts, C_u , β , b/B , and w are each placed on a horizontal (x) axis versus the obtained FOS (i.e., from Optum G2 analyses) on the vertical (y) axis. The FOS ranges within [0.8, 28.55] for all graphs. Not surprisingly, as a general view, the slope experienced more stability when a higher value of C_u was assigned to the soil (see Figure 2a). As Figure 2b,d illustrate, the parameters β (5°, 30°, 45°, 60°, and 75°) and w (50, 100, and 150 kN/m²) were inversely proportional to the FOS. This means the slope was more likely to fail when the model was performed with higher values of β and w . According to Figure 2c, the FOS did not show any significant sensitivity to the b/B ratio changes (0, 1, 2, 3, 4, and 5).

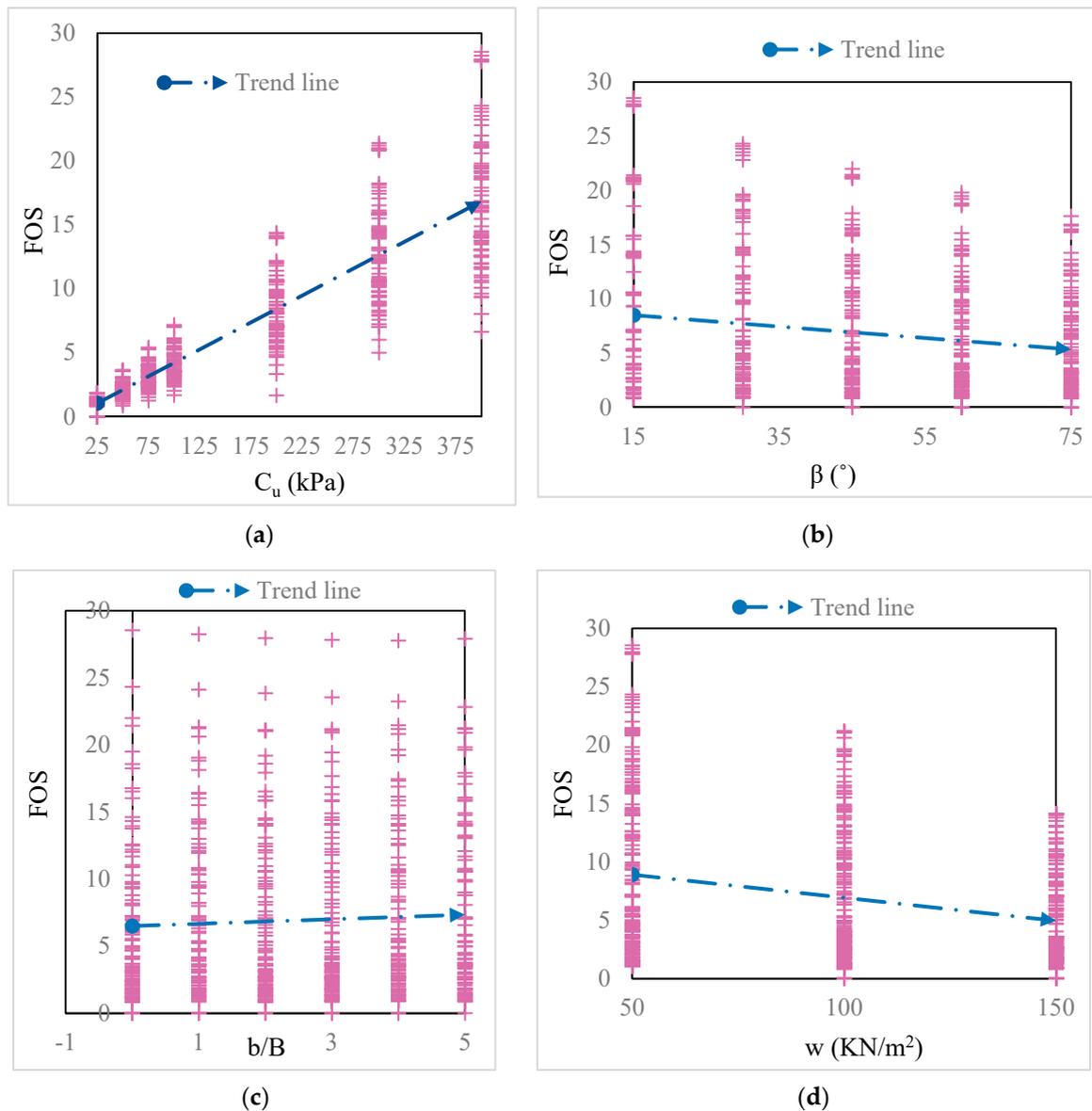


Figure 2. The relationship between the input parameters (a) strength of the soil (C_u), (b) slope angle (β), (c) setback distance ratio (b/B), and (d) applied surcharge on the footing installed over the slope (w) versus the obtained factor of safety (FOS).

Table 1 provides an example of the dataset used in this paper. The dataset was randomly divided into training and testing sub-groups, with the respective ratios of 0.8 (504 samples) and 0.2 (126 samples), based on previous studies [33]. Note that the training samples were used to train the MLR, MLP, RBF, SMO-SVM, IBK, RF, and RT models, and their performance was validated by means of the testing dataset. The 10-fold cross-validation process was used to select the training and test datasets (Figure 3). The results achieved were based on the best average accuracy for each classifier using 10-fold cross-validation (TFCV) based on the 20%/80% training/test split. In short, TFCV (which is a specific operation of k-fold cross-validation) divided our dataset into 10-partitions. In the first fold, a specific set of eight partitions of the data was used to train each model, whereas two partitions were used for testing. The accuracy for this fold was then indicated. This complied with using a various 20%/80% combinations of the data for training/testing, whose accuracy is also indicated. In the end, after all 10 folds were finished, the overall average accuracy was calculated [34,35].

Table 1. Example of inputs and output dataset used for training and validating the applied models.

No.	C_u	β	b/B	w	FOS	Stability Description	No.	C_u	β	b/B	w	FOS	Stability Description
1	25	15	0	50	1.872	Moderate	26	100	60	1	150	2.682	High
2	25	15	0	100	1.196	Low	27	100	60	2	50	4.671	Very high
3	25	15	0	150	0.8154	Unstable	28	100	60	2	100	3.674	High
4	25	15	1	50	1.852	Moderate	29	100	60	2	150	2.904	High
5	25	15	1	100	1.324	Low	30	100	60	3	50	4.72	Very high
6	25	15	1	150	0.8981	Unstable	31	100	60	3	100	3.753	High
7	25	15	2	50	1.834	Moderate	32	100	60	3	150	3.017	High
8	25	15	2	100	1.354	Low	33	100	60	4	50	4.833	Very high
9	25	15	2	150	0.9133	Unstable	34	100	60	4	100	3.904	High
10	25	15	3	50	1.824	Moderate	35	100	60	4	150	3.152	High
11	25	15	3	100	1.356	Low	36	100	60	5	50	4.996	Very high
12	25	15	3	150	0.9169	Unstable	37	100	60	5	100	4.059	Very high
13	25	15	4	50	1.825	Moderate	38	100	60	5	150	3.297	High
14	25	15	4	100	1.356	Low	39	100	75	0	50	4.227	Very high
15	25	15	4	150	0.9171	Unstable	40	100	75	0	100	2.425	High
16	25	15	5	50	1.832	Moderate	41	100	75	0	150	1.68	Moderate
17	25	15	5	100	1.357	Low	42	100	75	1	50	4.154	Very high
18	25	15	5	150	0.9179	Unstable	43	100	75	1	100	3.174	High
19	25	30	0	50	1.597	Moderate	44	100	75	1	150	2.344	High
20	25	30	0	100	1.035	Low	45	100	75	2	50	4.065	Very high
21	25	30	0	150	0.8	Unstable	46	100	75	2	100	3.185	High
22	25	30	1	50	1.581	Moderate	47	100	75	2	150	2.545	High
23	25	30	1	100	1.229	Low	48	100	75	3	50	4.116	Very high
24	25	30	1	150	0.8449	Unstable	49	100	75	3	100	3.272	High
25	25	30	2	50	1.556	Moderate	50	100	75	3	150	2.655	High

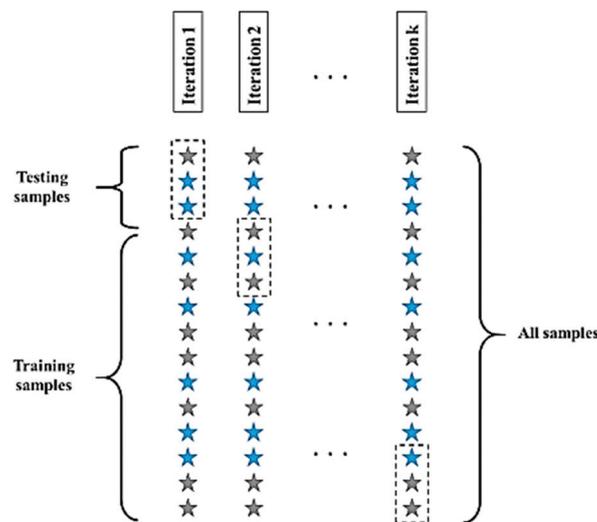


Figure 3. The k-fold cross-validation process.

In the following, the aim was to evaluate the competency of the applied models through a novel procedure based on the classification. In this sense, the actual values of the safety factor (i.e., obtained from the Optum G2 analysis) were stratified into five stability classes. The safety factor values varied from 0.8 to 28.55. Hence, the classification was carried out with respect to the extents given below (Table 2):

Table 2. Description of stability classification.

Class Number	Stability Description	FOS Range	Class Value
1	Unstable	(FOS < 1)	1
2	Low	(1 < FOS < 1.5)	2
3	Moderate	(1.5 < FOS < 2)	3
4	High	(2 < FOS < 4)	4
5	Very high	(4 < FOS)	5

As previously expressed, this study aimed to appraise the applicability of the seven most-employed machine learning models, namely multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT), toward evaluating the stability of a cohesive slope. For this purpose, the factor of safety (FOS) of the slope was estimated using the above-mentioned methods. Four slope stability effective factors that were chosen for this study were the undrained shear strength (C_u), slope angle (β), setback distance ratio (b/B), and applied surcharge on the footing installed over the slope (w). To prepare the input target dataset, 630 different stages were developed and analyzed in Optum G2 finite element software. As a result of this process, a FOS was derived for each performed stage. To create the required dataset, the values of C_u , β , b/B , and w were listed with the corresponding FOS. In the following, 80% of the prepared dataset (training phase consisting of 504 samples) was randomly selected for training the utilized models (MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT), and the effectiveness of them was evaluated by means of the remaining 20% of the dataset (testing phase consisting of 126 samples). The training process was carried out in WEKA software, which is an applicable tool for classification utilization and data mining. Note that many scholars have employed WEKA before for various simulating objectives [36,37]. To evaluate the efficiency of the implemented model, five statistical indices, namely the coefficient of determination (R^2), mean absolute error (MAE), root mean square error ($RMSE$), relative absolute error (RAE in %), and root relative squared error ($RRSE$ in %), were used to develop a color intensity ranking and present a visualized comparison of the results.

It should be noted that these criteria have been broadly used in earlier studies [38]. Equations (1)–(5) describe the formulation of R^2 , MAE , $RMSE$, RAE , and $RRSE$, respectively.

$$R^2 = 1 - \frac{\sum_{i=1}^s (Y_{i_{predicted}} - Y_{i_{observed}})^2}{\sum_{i=1}^s (Y_{i_{observed}} - \bar{Y}_{observed})^2} \quad (1)$$

$$MAE = \frac{1}{N} \sum_{i=1}^s |Y_{i_{observed}} - Y_{i_{predicted}}| \quad (2)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^s [(Y_{i_{observed}} - Y_{i_{predicted}})]^2} \quad (3)$$

$$RAE = \frac{\sum_{i=1}^s |Y_{i_{predicted}} - Y_{i_{observed}}|}{\sum_{i=1}^s |Y_{i_{observed}} - \bar{Y}_{observed}|} \quad (4)$$

$$RRSE = \sqrt{\frac{\sum_{i=1}^s (Y_{i_{predicted}} - Y_{i_{observed}})^2}{\sum_{i=1}^s (Y_{i_{observed}} - \bar{Y}_{observed})^2}} \quad (5)$$

In all the above equations, $Y_{i_{observed}}$ and $Y_{i_{predicted}}$ stand for the actual and predicted values of the FOS, respectively. The term S is the indicator of the number of data points and $\bar{Y}_{observed}$ is the average of the actual values of the FOS. The outcomes of this paper are presented in various ways. In the next part, the competency of the applied models (i.e., MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT) for the approximation of FOS is extensively presented, compared, and discussed.

2.1. Machine Learning Techniques

The present research outlines the application of various machine-learning-based predictive models, namely multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT), in slope stability assessment. Analysis was undertaken using WEKA software [39]. The mentioned models are described below.

2.1.1. Multiple Linear Regression (MLR)

Fitting a linear equation to the data samples (Figure 4) is the essence of the multiple linear regression (MLR) model. It tries to reveal the generic equation between two or more explanatory (independent) variables and a response (dependent) variable. Equation (6) defines the general equation of the MLR [40]:

$$y = \rho_0 + \rho_1 x_1 + \rho_2 x_2 + \dots + \rho_s x_s + \chi \quad (6)$$

where x and y are the independent and dependent variables, respectively. The terms $\rho_0, \rho_1, \dots, \rho_s$ indicate the unknown parameters of the MLR. Also, χ is the random variable in the MLR generic formula that has a normal distribution.

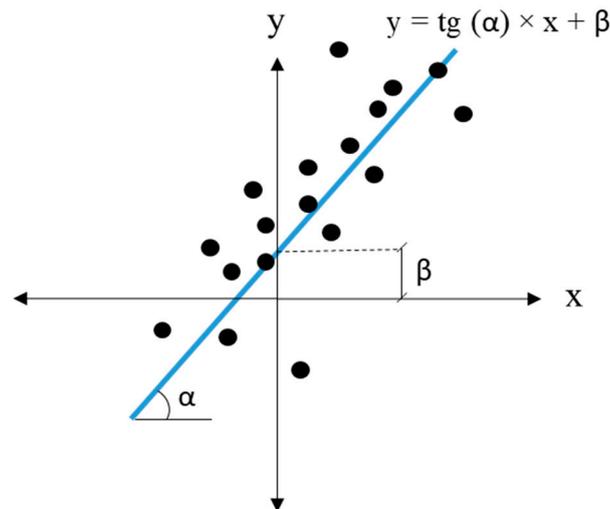


Figure 4. A simple view of linear fitting.

The main task of MLR is to forecast the unknown terms (i.e., $\rho_0, \rho_1, \dots, \rho_s$) of Equation (6). By applying the least-squares technique, the practical form of the statistical regression method is represented as follows [40]:

$$y = p_0 + p_1x_1 + p_2x_2 + \dots + p_sx_s + \mu \quad (7)$$

In the above equation p_0, p_1, \dots, p_s are the approximated regression coefficients of $\rho_0, \rho_1, \dots, \rho_s$, respectively. Also, the term μ gives the estimated error for the sample. Assuming the μ is the difference between the real and modelled y , the estimation of y is performed as follows

$$\hat{y} = p_0 + p_1x_1 + p_2x_2 + \dots + p_sx_s \quad (8)$$

2.1.2. Multi-Layer Perceptron (MLP)

Imitating the human neural network, an artificial neural network (ANN) was first suggested by McCulloch and Pitts [41]. The non-linear rules between each set of inputs and their specific target(s) can be easily revealed using ANN [42]. Many scholars have used different types of ANN to model various engineering models [14,16,20,38,43,44]. Despite the explicit benefits of ANNs, the most significant disadvantages of them lie in being trapped in their local minima and overfitting. A highly connected pattern forms the structure of an ANN. A multi-layer perceptron is one of the most prevalent types of this model. A minimum of three layers is needed to form the MLP structure. Every layer contains a different number of so-called computational nodes called “neurons.” Figure 5a illustrates the schematic view of the MLP neural network that was used in this study. This network had two hidden layers comprising 10 neurons each. Every line in this structure is indicative of an interconnection weight (W). A more detailed view of the performance of a typical neuron is shown in Figure 5b:

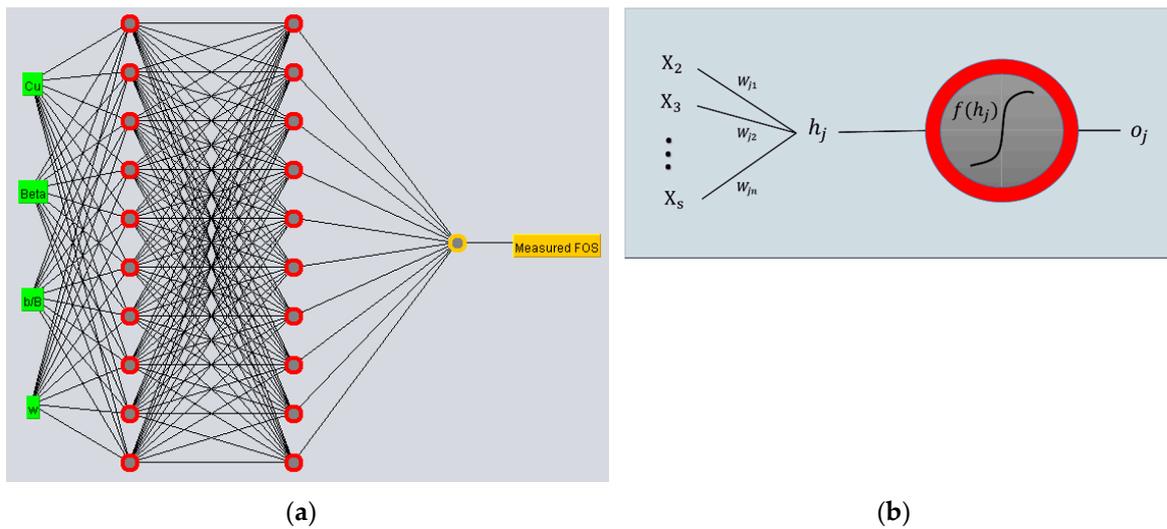


Figure 5. The schematic view of (a) the proposed model for the multilayer perceptron and (b) the operation of computational neurons.

Note that, in Figure 1b, the parameter h_j is calculated as follows:

$$h_j = \sum_{i=1}^s W_{ji}X_i + b_j \tag{9}$$

where the terms W , X , and b stand for the weight, input, and bias of the utilized neuron. Also, the function f denotes the activation function of the neuron that is applied to produce the output.

2.1.3. Radial Basis Function Regression (RBFR)

The radial basis function (RBF) approach (shown in Figure 6) was first presented by Broomhead and Lowe [45] for exact function interpolation [46]. The RBF regression model attempts to fit a set of local kernels in high-dimensional space to solve problems [45,47,48]. The position of noisy instances is considered for executing the fitting. The number of purposed kernels is often estimated using a bath algorithm.

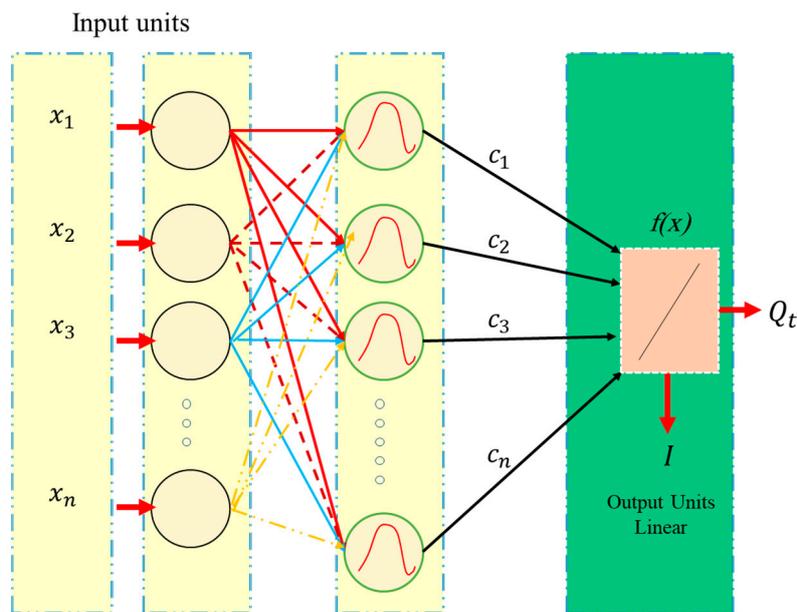


Figure 6. The structure of Radial basis function.

The RBF regression performs through establishing an expectation function with the overall form of Equation (10):

$$F(x) = \sum_{i=1}^z k(\|x - x_i\|) r_i \tag{10}$$

where $\|x\|$ represents the Euclidean norm on x and $\{k(\|x - x_i\|) | i = 1, 2, \dots, m\}$ is a set of m RBFs, which are characterized by their non-linearity and being constant on x . Also, the regression coefficient is denoted by r_i in this formula. Every training sample (i.e., x_i) determines a center of the RBFs. In other words, the so-called function “radial” is impressively dependent on the distance (ψ) between the center (x_i) and the origin (x) (i.e., $\psi = \|x - x_i\|$). The Gaussian $k(r) = \exp(-\sigma\psi^2)$, for $\psi \geq 0$ [49,50]; the multi-quadric $k(r) = \sqrt{\delta^2 + \psi^2}$, for $\delta \geq 0$ [51]; and the thin-plate-spline $k(r) = \psi^2 \log \psi$ [52,53] functions are three extensively applied examples of the function $k(x)$ following the RBF rules.

2.1.4. Improved Support Vector Machine using Sequential Minimal Optimization Algorithm (SMO-SVM)

Support Vector Machine (SVM)

Insulating different classes is the pivotal norm during the performance of the support vector machine (SVM). This happens through detecting the decision boundaries. SVMs have been shown to have a suitable accuracy in various classification problems when dealing with high-dimensional and non-separable datasets [54–56]. Theoretically, SVMs act according to statistical theories [57] and aim to transform the problem from non-linear to linear feature spaces [58]. More specifically, assessing the support vectors that are set at the edge of the module descriptors will lead to locating the most appropriate hyperplane between the modules. Note that guiding cases that could not be a support vector are known as neglected vectors [59]. The graphical description of the mentioned procedure is depicted in Figure 7:

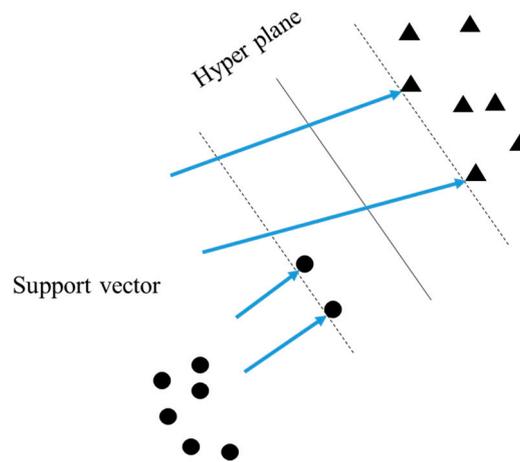


Figure 7. The graphical depiction of the mechanism of an SVM.

In the SVMs, the formulation of the hyperplane (i.e., the decision surface) is as follows:

$$\sum_{j=1}^V l_j \cdot f_j \cdot K(x, x_j) = 0 \tag{11}$$

where l_j , x , and f_j represent the Lagrange coefficient, the vector that is drawn from the input space, and the j th output, respectively. Also, $K(x, x_j)$ is indicative of the inner-product kernel (i.e., the linear kernel function) and operates as given in Equation (12):

$$K(x, x_j) = x \cdot x_j \tag{12}$$

where x and x_j stand for the input vector and input pattern related to the j th instance, respectively.

The vector w and the biasing threshold b are obtained from Equations (13) and (14) [60]:

$$w = \sum_{i=1}^V l_i \cdot f_i \cdot x_i \tag{13}$$

$$b = w \cdot x_k - f_k \tag{14}$$

Also, $Q(l)$, which is a quadratic function (in l_i), is given using the following formula:

$$\begin{aligned} \min Q(l) &= \min \frac{1}{2} \sum_{j=1}^V \sum_{h=1}^V f_j f_h K(x_j, x_h) l_j l_h - \sum_{j=1}^V l_j \\ \text{where,} & \\ 0 &\leq l_j \leq C \\ \sum_{j=1}^V l_j \cdot f_j &= 0 \end{aligned} \tag{15}$$

Sequential Minimal Optimization (SMO)

Sequential minimal optimization (SMO) is a simple algorithm that was used in this study to improve the performance of the SVM. When corresponding the linear weight vector to the optimal values of l_i using SMO, adding a matrix repository and establishing a repetitive statistical is not required. Every pair of Lagrange multiplier, such as l_1 and l_2 ($l_1 < l_2$), vary from 0 to C. During the SMO performance, the l_2 is first computed. The extent of the transverse contour fragment is determined based on the obtained l_2 [60]. If the targets f_1 and f_2 are equal, the subsequent bounds apply to l_1

(Equation (10)); otherwise, they apply to l_2 . Equation (16) defines the directional minimum along with the contour fragment acquired using SMO:

$$l_2' = l_2 + \frac{f_2(E_1 - E_2)}{\lambda} \tag{16}$$

where E_j is the difference between the j th SMV output and the target. The parameter λ stands for an unbiased function and is given by the following equation:

$$\lambda = K(x_1, x_1) + K(x_2, x_2) - 2K(x_1, x_2) \tag{17}$$

In the following, the l_2' clipped to the range $[A, B]$:

$$A = \max(0, l_2 - l_1) \quad B = \min(C, C + l_2 - l_1) \tag{18}$$

$$A = \max(0, l_2 + l_1 - C) \quad B = \min(C, l_2 + l_1) \tag{19}$$

$$l_2'' = \begin{cases} B & \text{if } l_2' \geq B \\ l_2' & \text{if } A < l_2' < B \\ A & \text{if } l_2' \leq A \end{cases} \tag{20}$$

$$l_1' = l_1 + v(l_2 - l_2''), \quad v = f_1 f_2 \tag{21}$$

Finally, l_1 and l_2 are moved to the bound with the lowest unbiased function.

After optimizing the Lagrange multipliers, SVM is eligible to give the output f_1 using the threshold b_1 , if $0 < l_1 < C$ [60]. Equations (22) and (23) show the way that b_1 and b_2 are calculated:

$$b_1 = E_1 + f_1(l_1' - l_1)K(x_1, x_1) + f_2(l_1'' - l_2)K(x_1, x_2) + b \tag{22}$$

$$b_2 = E_1 + f_1(l_1' - l_1)K(x_1, x_2) + f_2(l_1'' - l_2)K(x_2, x_2) + b \tag{23}$$

2.1.5. Lazy k-Nearest Neighbor (IBK)

In the lazy learning procedure, the generalization of the training sample commences with producing a query and learners do not tend to operate until the classification time. Generating the objective function through a local estimation can be stated as the main advantage of the models associated with lazy learning. One of the most prevalent techniques that follow these rules is the k-nearest neighbor algorithm [61]. Multiple problems can be simulated with this approach due to the local estimation accomplished for all queries to the system. Moreover, this feature causes the capability of such systems to work, even when the problem conditions change [62,63]. Even though lazy learning has been shown to have a suitable quickness for training, validation takes place over longer time. Furthermore, since the complete training process is stored in this method, a large space is required for the k-nearest neighborhood (IBK) classifier as a non-parametric model that has a good background for classification and regression utilization [61]. The number of nearest neighbors can be specified exactly in the object editor or set automatically through applying cross-validation to a certain upper bound. IBK utilizes various search algorithms (that are linear by default) to expedite finding the nearest neighbors. Also, the default function for evaluating the position of samples is the Euclidean distance, which can be substituted by Minkowski, Manhattan, and Chebyshev distance functions [64]. As is well-discussed in References [62,65], the distance from the validation data can be a valid criterion to weight the estimations from more than one neighbor. Note that altering the oldest training data with new possible samples takes place by determining a window size. It helps to have a constant number of samples [66].

2.1.6. Random Forest (RF)

The random forest (RF) is a capable tool used for classification problems [67]. RF follows the ensemble learning rules by employing several classification trees [68]. In this model, showing better results than entire individual trees (i.e., the better performance of an ensemble of classification trees) entails a greater than random accuracy, as well as the variety of the ensemble members [69]. During the FR implementation, it aims to change the predictive relations randomly to increase the diversity of trees in the utilized forest. The development of the forest necessitates the user to specify the values of the number of trees (t) and the number of variables that cleave the nodes (g). Note that either categorical or numerical variables are eligible for this task. The parameter t needs to be high enough to guarantee the reliability of the model. The unbiased estimation of the generalization error (GE), called the out-of-bag (OOB) error is obtained, when the random forest is being formed. Breiman [67] showed that a limiting value of GE is produced by RF. The OOB error is computed using the ratio of the misclassification error over the total out-of-bag elements. RF attempts to appraise the prominence of the predictive variables through considering the increase of the OOB error. This increment is caused by permuting the OOB data for the utilized variable, while other variables are left without any change. It must be noted that the prominence of the predictive variable rises as the OOB error increases [70]. Since each tree is counted as a completely independent random case, the affluence of random trees can lead to reducing the likelihood of overfitting. Note that this can be considered a key advantage of RF. This algorithm can automatically handle the missing values due to its suitable resistance against the outliers when making predictions.

2.1.7. Random Tree (RT)

Random trees (RT) symbolize a supervised classification model first introduced by Leo Breiman and Adele Cutler [67]. Similar to RF, ensemble learning forms the essence of RT. In this approach, many learners who operate individually are involved. To develop a decision tree, a bagging idea is used to provide a randomly chosen group of samples. The pivotal difference between a standard tree and random forest (FR) is found in the splitting of the node. In RF, this is executed using the best among the subgroup of predictors, while a standard tree uses the elite split among all variables. The RT is an ensemble of tree forecasters (i.e., forest) and can handle both regression and classification utilizations. During the RT execution, the input data are given to the tree classifier. Every existing tree performs the classification of the inputs. Eventually, the category with the highest frequency is the output of the system. This is noteworthy because the training error is calculated internally; none of cross-validation and bootstraps are needed to estimate the accuracy of the training stage. Note that the average of the responses of the whole forest members is computed to produce the output of the regression problems [71]. The error of this scheme is a proportion of the number of misclassified vectors to all vectors in the original data.

3. Results and Discussion

The calculated values of R^2 , MAE , $RMSE$, RAE , and $RRSE$ for estimating the FOS are tabulated in Tables 3 and 4 for the training and testing datasets. Also, the total ranking obtained for the models is featured in Table 5. A color intensity model is given to denote the quality of results graphically. A red collection is considered for particular results. A higher value of R^2 and lower MAE , $RMSE$, RAE , and $RRSE$ are represented using a more intense red color. In the last column, the intensity of a green collection is supposed to be inversely proportional to the ranking gained by each model.

Table 3. Total ranking of the training dataset in predicting the factor of safety.

Proposed Models	Network Results					Ranking the Predicted Models					Total Ranking Score	Rank
	R ²	MAE	RMSE	RAE (%)	RRSE (%)	R ²	MAE	RMSE	RAE (%)	RRSE (%)		
MLR	0.9586	1.2527	1.7366	25.0515	28.4887	2	1	2	1	2	8	6
MLP	0.9937	0.494	0.7131	9.8796	11.6985	3	3	3	3	3	15	5
RBFR	0.9948	0.3976	0.6231	7.9522	10.2221	4	4	4	4	4	20	4
SMO-SVM	0.9529	1.161	1.9183	23.2182	31.4703	1	2	1	2	1	7	7
IBK	1.0000	0	0	0	0	7	7	7	7	7	35	1
RF	0.9997	0.1063	0.1486	2.1254	2.4385	5	5	5	5	5	25	3
RT	0.9998	0.0811	0.1111	1.622	1.8221	6	6	6	6	6	30	2

Note: It is a ranking system based on colures. The one with higher intensity has better value. R²: correlation coefficient; MAE: mean absolute error; RMSE: root mean squared error; RAE: relative absolute error; RRSE: root relative squared error.

Table 4. Total ranking of the testing dataset in predicting the factor of safety.

Proposed Models	Network Results					Ranking the Predicted Models					Total Ranking Score	Rank
	R ²	MAE	RMSE	RAE (%)	RRSE (%)	R ²	MAE	RMSE	RAE (%)	RRSE (%)		
MLR	0.9649	1.1939	1.5891	24.1272	26.4613	1	1	2	1	2	7	7
MLP	0.9939	0.5149	0.7093	10.4047	11.8116	4	4	4	4	4	20	4
RBFR	0.9955	0.3936	0.5667	7.9549	9.4376	6	5	6	5	6	28	2
SMO-SVM	0.9653	1.0360	1.6362	20.9366	27.247	2	2	1	2	1	8	6
IBK	0.9837	0.8184	1.1066	16.5388	17.247	3	3	3	3	3	15	5
RF	0.9985	0.2152	0.3312	4.3498	5.5145	7	7	7	7	7	35	1
RT	0.9950	0.3492	0.6033	7.0576	10.047	5	6	5	6	5	27	3

Table 5. Total ranking of both training and testing datasets in predicting the factor of safety.

Proposed Model	Network Result										Total Score	Total Rank
	Training Dataset					Testing Dataset						
	R ²	MAE	RMSE	RAE (%)	RRSE (%)	R ²	MAE	RMSE	RAE	RRSE		
MLR	2	1	2	1	2	1	1	2	1	2	15	6
MLP	3	3	3	3	3	4	4	4	4	4	35	5
RBFR	4	4	4	4	4	6	5	6	5	6	48	4
SMO-SVM	1	2	1	2	1	2	2	1	2	1	15	6
IBK	7	7	7	7	7	3	3	3	3	3	50	3
RF	5	5	5	5	5	7	7	7	7	7	60	1
RT	6	6	6	6	6	5	6	5	6	5	57	2

At a glance, RF could be seen to be the outstanding model due to its highest score. However, based on the acquired R² for the training (0.9586, 0.9937, 0.9948, 0.9529, 1.000, 0.9997, and 0.9998, calculated for MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT, respectively) and testing (0.9649, 0.9939, 0.9955, 0.9653, 0.9837, 0.9985, and 0.9950, respectively) datasets, an acceptable correlation could be seen for all models. Referring to the RMSE (28.4887, 1.7366, 0.7131, 0.6231, 1.9183, 0.00, 0.1486, and 0.1111) and RRSE (11.6985%, 10.2221%, 31.4703%, 0.0000%, 2.4385%, and 1.8221%) obtained for the training dataset, the IBK predictive model presented the most effective training compared to other models. Also, RT and RF could be seen as the second- and third-most accurate models, respectively, in the training stage. In addition, RBFR outperformed MLP, MLR, and SMO-SVM (as shown in Table 3).

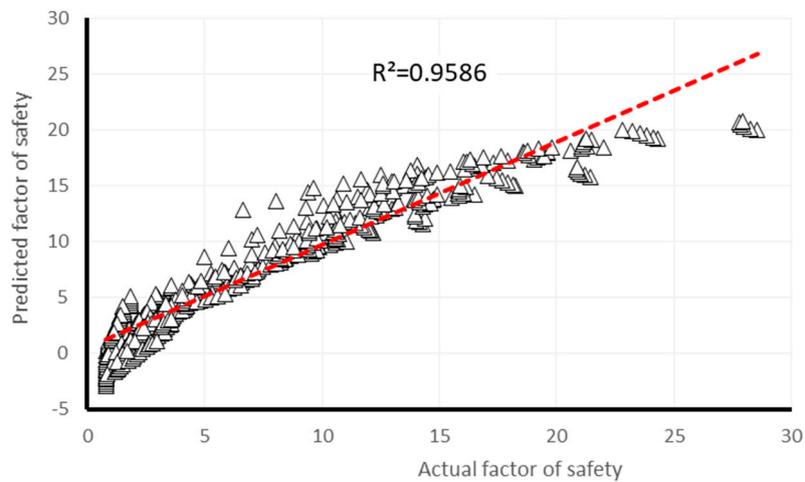
The results for evaluating the performance of the models used are available in Table 4. Despite this fact that IBK showed an excellent performance during the training phase, a better validation was achieved using the RF, RBFR, RT, and MLP methods in this part. This claim can be proved by the obtained values of R² (0.9649, 0.9939, 0.9955, 0.9653, 0.9837, 0.9985, and 0.9950), MAE (1.1939, 0.5149, 0.3936, 1.0360, 0.8184, 0.2152, and 0.3492) and RAE (24.1272%, 10.4047%, 7.9549%, 20.9366%, 16.5388%, 4.3498%, and 7.0576%) for MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT, respectively. Respective values of 0.9985, 0.2152, 0.3312, 4.3498%, and 5.5145% computed for R², MAE, RMSE, RAE, and RRSE indices show that the highest level of accuracy in the validation phase was achieved by RF. Similar to the training phase, the lowest reliability was achieved by the SVM and MLR models, compared to the other utilized techniques.

Furthermore, a comprehensive comparison (regarding the results of both the training and testing datasets) of applied methods is conducted in Table 5. In this table, considering the assumption of individual ranks obtained for each model (based on the R², MAE, RMSE, RAE, and RRSE in Tables 3 and 4), a new total ranking is provided. According to this table, the RF (total score = 60) was the most efficient model in the current study due to its good performance in both the training and testing phases. After RF, RT (total score = 57) was the second-most reliable model. Other employed methods including IBK (total score = 50), RBFR (total score = 48), and MLP (total score = 35) were the third-, fourth-, and fifth-most accurate tools, respectively. Also, the poorest estimation was given by the SMO-SVM and MLR approaches (total score = 15 for both models).

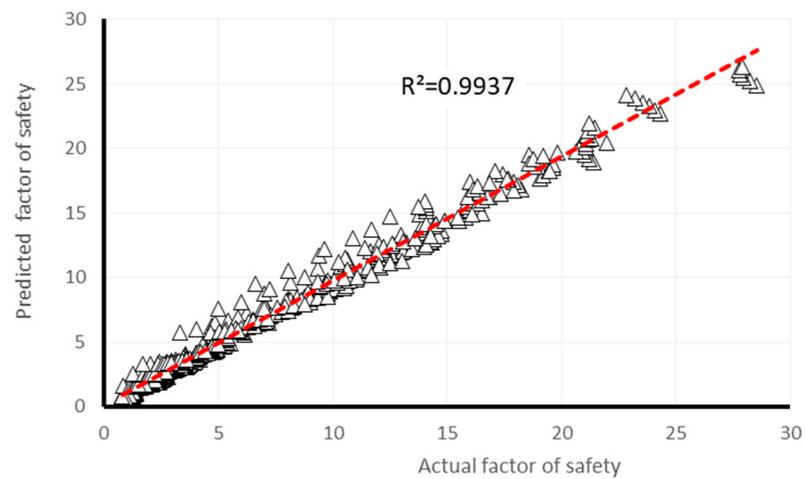
Application of multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT) was found to be viable for appraising the stability of a single-layered cohesive slope. The models that use ensemble learning (i.e., RF and RT) outperformed the models that are based on lazy-learning (IBK), regression rules (MLR and RBFR), neural learning (MLP), and statistical theories (SMO-SVM). Referring to the acquired total scores (the summation of scores obtained for the training and testing stages) of 15, 35, 48, 15, 50, 60, and 57 for MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT, respectively, shows the superiority of the RT algorithm. After that, RT, IBK, and RBFR were shown to have an excellent performance when estimating the FOS. Also, satisfying reliability was found for the MLP, SMO-SVM, and MLR methods. Based on the results obtained for the classification of the risk of failure,

the closest approximation was achieved using the RF, RT, IBK, and SMO-SVM methods. In this regard, 14.60% of the dataset was assorted as hazardous slopes (i.e., “unstable” and “low stability” categories). This value was estimated as 18.25%, 12.54%, 12.70%, 14.76%, 14.60%, 13.81%, and 14.13% by the MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT models, respectively.

It is well established that the best relationship between the data shown on the horizontal and vertical axis is demonstrated using the line $x = y$, in the regression chart. According to Figures 8 and 9, the IBK produced the closest outputs to the actual values of the safety factor in the training phase ($R^2 = 1$). Also, the highest correlation was achieved by the RF outputs for the testing data ($R^2 = 0.9985$). A higher scattering in samples, observed for MLR ($R^2 = 0.9586$ and 0.9649) and SMO-SVM ($R^2 = 0.9529$ and 0.9653) predictions indicates a lower sensitivity of these models in both the training and testing datasets.

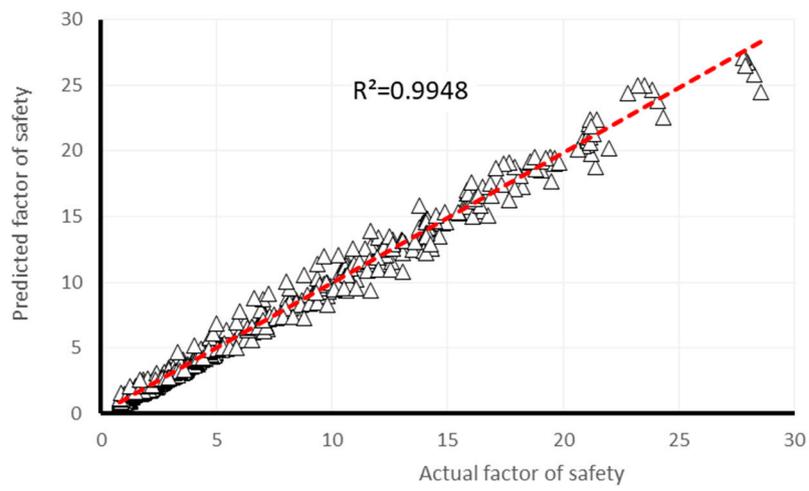


a

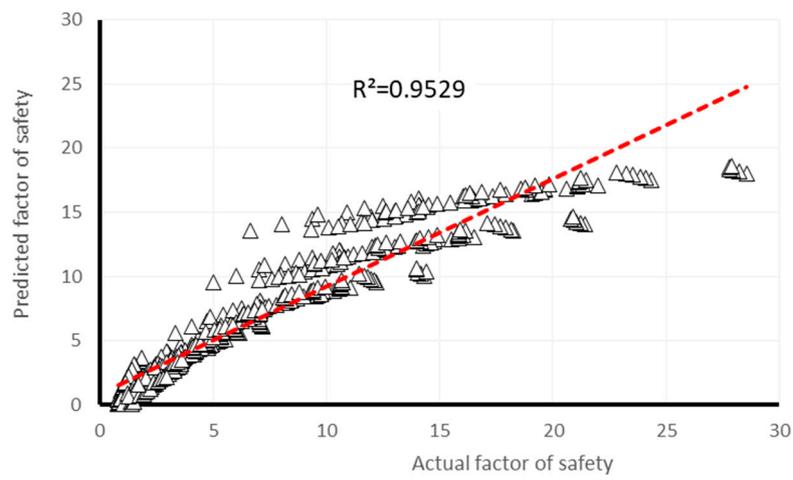


b

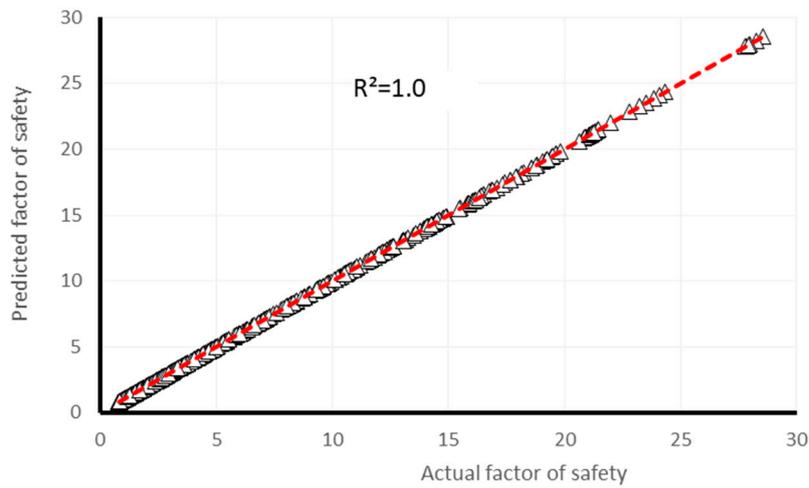
Figure 8. Cont.



c



d



e

Figure 8. Cont.

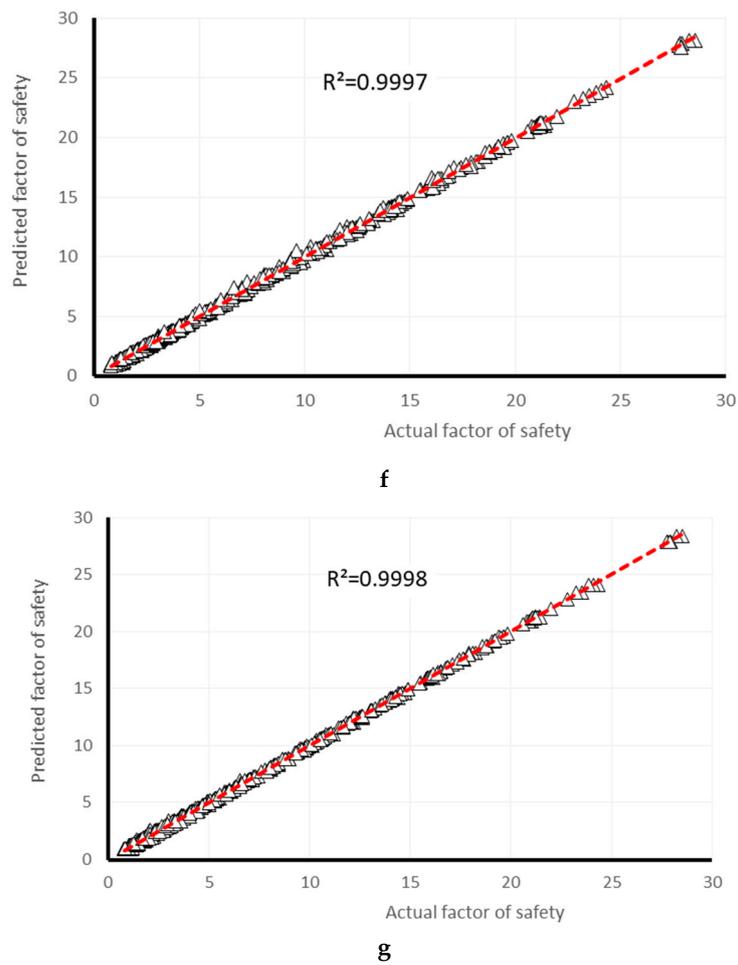


Figure 8. The network result for the training dataset in (a) multiple linear regression (MLR), (b) multi-layer perceptron (MLP), (c) radial basis function regression (RBFR), (d) improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), (e) lazy k-nearest neighbor (IBK), (f) random forest (RF), and (g) random tree (RT).

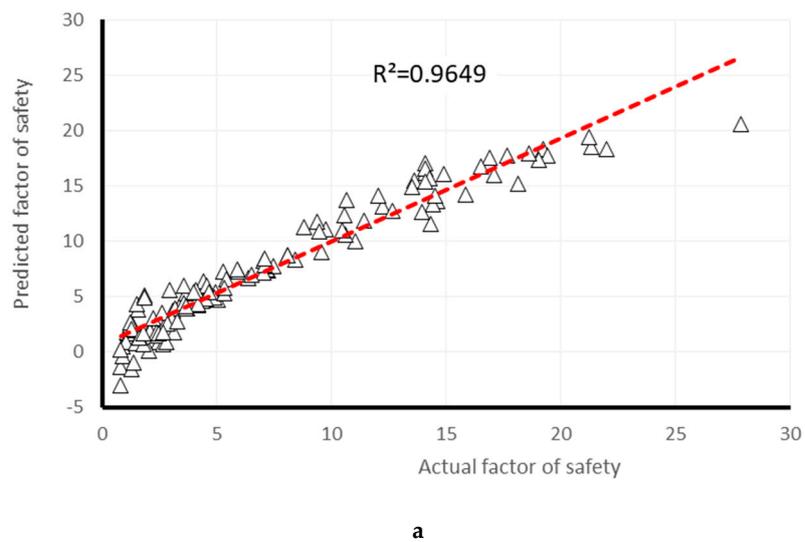
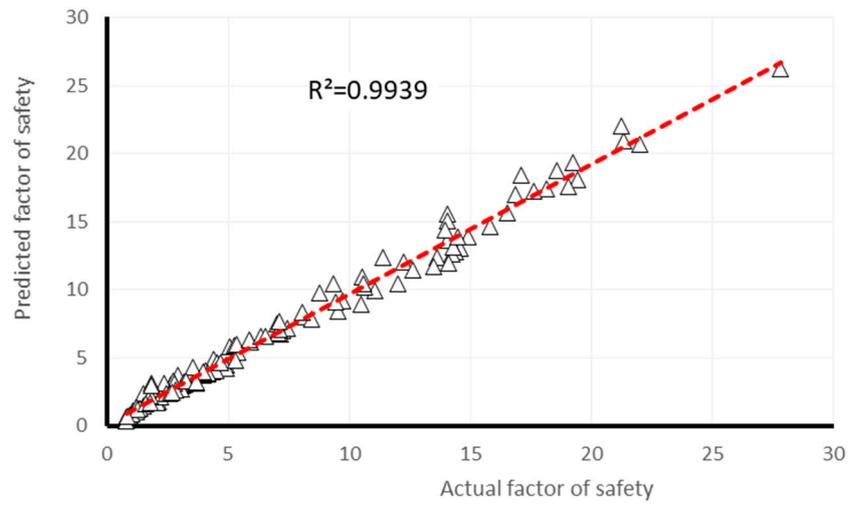
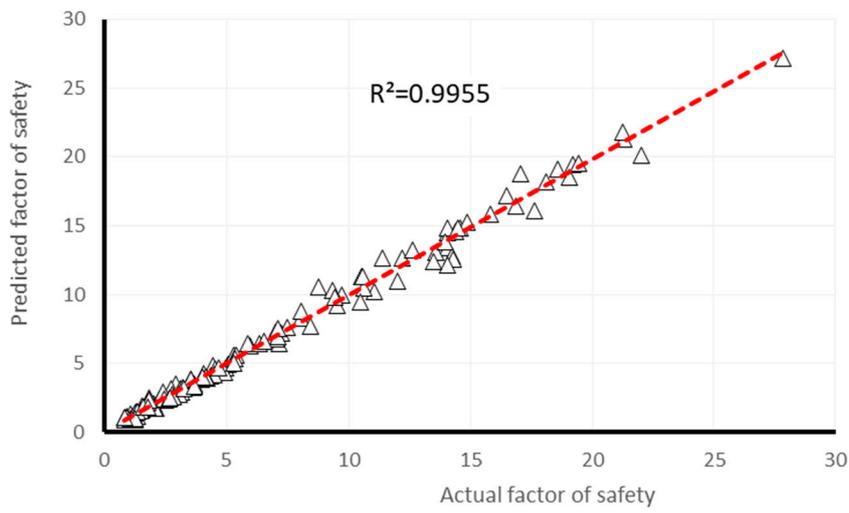


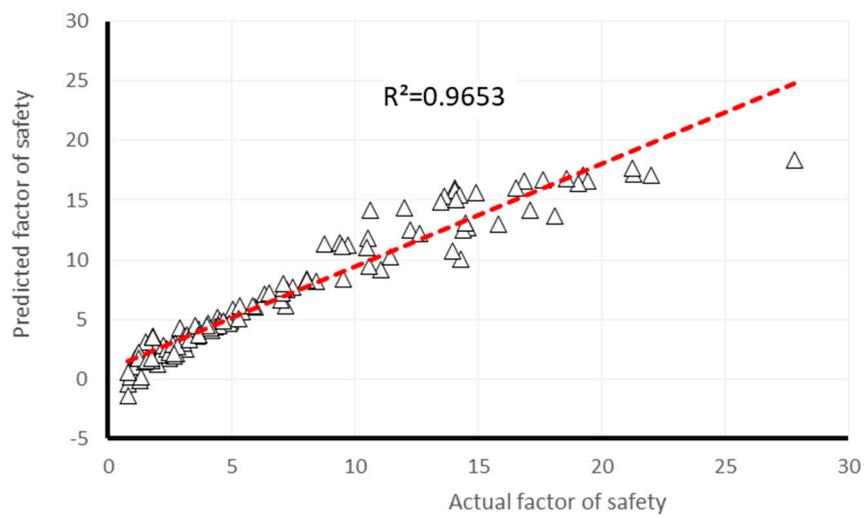
Figure 9. Cont.



b

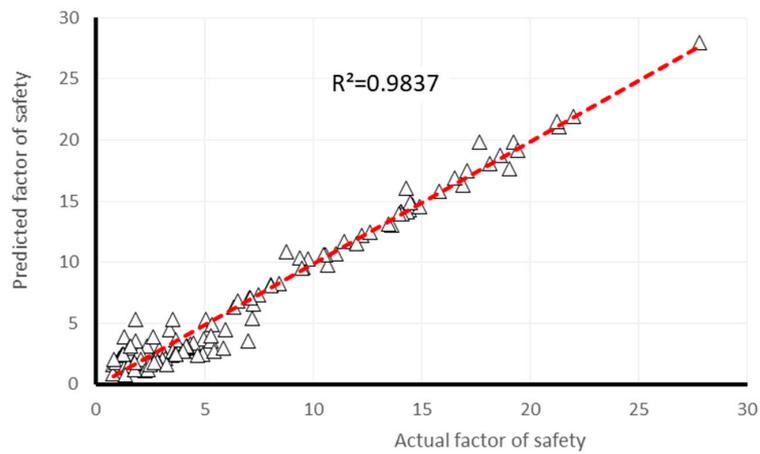


c

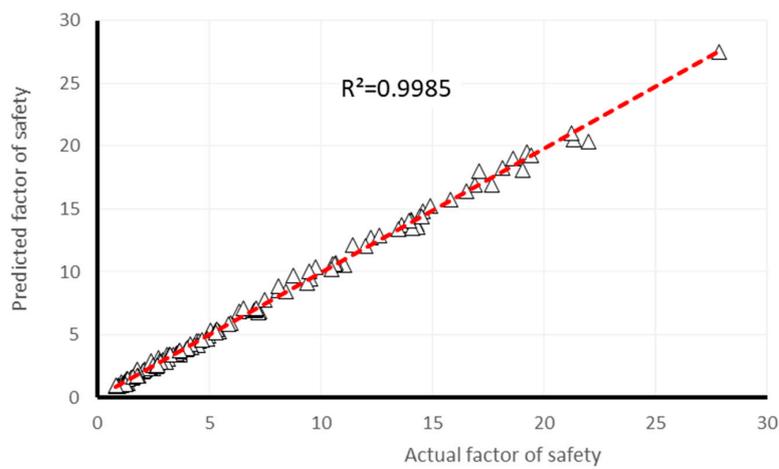


d

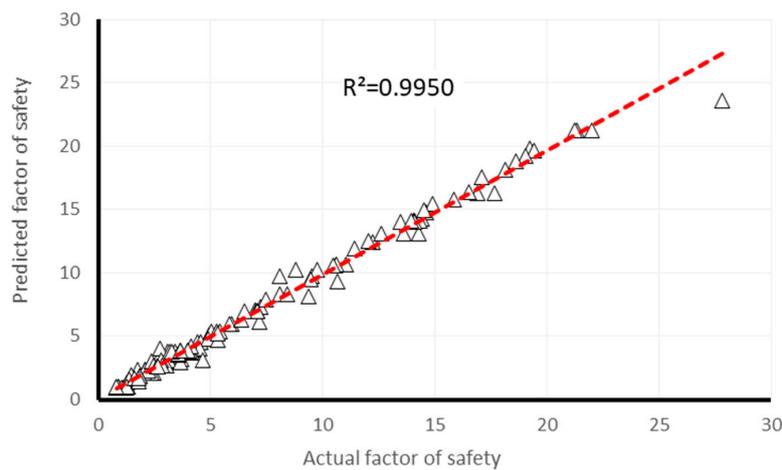
Figure 9. Cont.



e



f



g

Figure 9. The network result for the testing dataset in (a) multiple linear regression (MLR), (b) multi-layer perceptron (MLP), (c) radial basis function regression (RBFR), (d) improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), (e) lazy k-nearest neighbor (IBK), (f) random forest (RF), and (g) random tree (RT).

The percentage of each stability class (i.e., the number of safety factor values in the utilized class over the whole number of data points) was calculated. The results of this part are depicted in the form of a column chart in Figure 10.

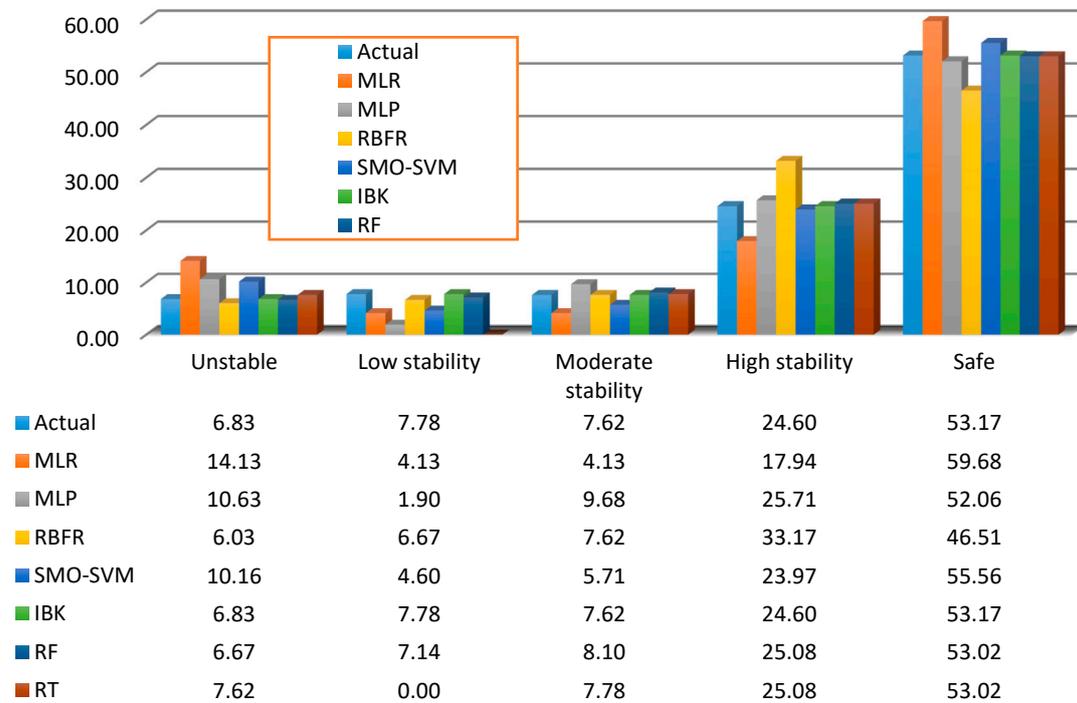


Figure 10. The column chart showing the percentage of stability classes.

In addition, the regression between the actual and estimated values of FOS is depicted in Figures 6 and 7 for the training and testing data.

Based on this chart, more than half of the safety factor values indicated a safe condition for the examined slope. Almost all employed models showed a good approximation for the classification of the risk of failure. Considering the categories “unstable” and “low stability” as the dangerous situation of the slope (i.e., the slope was more likely to fail with these safety factors), a total of 14.60% of the dataset was assorted as being dangerous. This value was estimated as 18.25%, 12.54%, 12.70%, 14.76%, 14.60%, 13.81%, and 14.13% by the MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT models, respectively.

$$FOS_{MLR} = 0.042 \times C_u - 0.0525 \times \beta + 0.1718 \times b/B - 0.0395 \times w + 5.9289 \quad (24)$$

$$FOS_{SMO-SVM} = + 0.5385 \times C_u - 0.0685 \times \beta + 0.0213 \times b/B - 0.0925 \times w + 0.0825 \quad (25)$$

$$FOS_{MLP} = -0.2956 \times Z_1 + 0.0456 \times Z_2 + 0.9411 \times Z_3 + 1.3988 \times Z_4 + 0.1028 \times Z_5 - 0.0499 \times Z_6 + 0.0303 \times Z_7 - 0.1266 \times Z_8 - 0.7543 \times Z_9 - 0.0020 \times Z_{10} - 1.0782 \quad (26)$$

where the parameters Z_1, Z_2, \dots, Z_{10} are shown in Table 6 and weight and biases tabulated in Table 7:

Table 6. Weight and biases of the ANN model.

Neuron (i)	$Z_i = \text{Tansig}(W_{i1} \times Y_1 + W_{i2} \times Y_2 + \dots + W_{i10} \times Y_{10} + b_i)$										
	W_{i1}	W_{i2}	W_{i3}	W_{i4}	W_{i5}	W_{i6}	W_{i7}	W_{i8}	W_{i9}	W_{i10}	b_i
1	0.8014	0.2141	-0.9739	0.3018	-1.0577	0.3676	0.0819	0.0814	0.1832	-0.2797	-0.1420
2	0.7150	-1.0819	2.6119	1.4173	-1.7753	-0.6882	-0.3629	-0.3758	0.3733	3.8024	-0.8470
3	0.0539	-0.4013	-0.0124	-0.1376	1.5404	-0.1659	-0.1569	0.0965	0.1123	-1.0206	0.7256
4	0.0975	0.2092	-0.2620	-0.0167	-0.2081	0.1904	0.0662	0.0342	-0.0256	1.3529	0.6269
5	0.3407	-0.2303	0.6002	0.7437	-0.0743	0.2708	0.6381	-0.4040	-0.2603	1.6380	0.0664
6	0.2808	-1.1512	1.6090	0.0751	-0.9023	1.0409	-0.1826	1.3060	-0.0688	-0.6791	-0.8171
7	-1.4209	-1.3960	-0.6175	0.0115	0.4415	1.3083	-0.3546	1.1381	-0.5560	-1.3212	-0.4100
8	-0.8622	-1.4606	1.0247	-1.4288	-2.0929	0.8496	0.1114	0.1647	-0.4303	-0.4968	-0.1472
9	0.0532	-0.0747	-0.5253	0.0773	-1.0538	-0.1710	-0.1178	-0.0304	0.0095	-0.7900	-1.2037
10	1.6879	-1.7526	-0.5477	2.4484	-0.1138	-1.9688	0.5682	2.5085	-4.1707	-0.5766	-0.5541

where the parameters Y_1, Y_2, \dots, Y_{10} are in Table 7.

Table 7. Weight and biases of the ANN model.

Neuron (i)	$Y_i = \text{Tansig}(W_{i1} \times C_u + W_{i2} \times \beta + W_{i10} \times (b/B) + W_{i2} \times w + b_i)$				
	W_{i1}	W_{i2}	W_{i3}	W_{i4}	b_i
1	1.1179	2.1216	-0.9307	-1.9126	-3.1819
2	-1.6540	0.7670	0.5226	0.5905	1.9420
3	-1.7577	0.1682	1.0751	-0.8448	1.5533
4	-1.1786	-1.2008	1.5102	-1.4182	1.2264
5	0.3132	0.0403	0.0188	-0.0171	-0.0729
6	0.3103	-0.1702	0.8267	-2.1844	0.1736
7	-0.2284	-1.6277	1.7207	1.2254	-0.0037
8	1.9007	-1.8777	0.9780	2.9271	1.4079
9	0.1767	-1.2933	-1.2124	-0.5631	1.4512
10	0.5156	-0.3139	-0.0902	-0.3256	-1.2137

C_u —undrained cohesion strength (kPa), β —slope angle ($^\circ$), b/B —setback distance ratio, and w —applied surcharge on the footing.

4. Conclusions

In the past, many scholars have investigated the slope failure phenomenon due to its huge impact on many civil engineering projects. Due to this, many traditional techniques offering numerical and finite element solutions have been developed. The appearance of machine learning has made these techniques antiquated. Therefore, the main incentive of this study was to evaluate the proficiency of various machine-learning-based models, namely multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT), in estimating the factor of safety (FOS) of a cohesive slope. To provide the required dataset. Optum G2 software was effectively used to calculate the FOS of 630 different slope conditions. The inputs (i.e., the slope failure conditioning factors) chosen were the undrained shear strength (C_u), slope angle (β), setback distance ratio (b/B), and applied surcharge on the shallow foundation installed over the slope (w). To train the intelligent models, 80% of the provided dataset (i.e., 504 samples) was randomly selected. Then, the results of each model were validated using the remaining 20% (i.e., 126 samples). Five well-known statistical indices, namely coefficient of determination (R^2), mean absolute error (MAE), root mean square error (RMSE), relative absolute error (RAE in %), and root relative squared error (RRSE in %), were used to evaluate the results of the MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT predictive models. A color intensity rating along with the total ranking method (i.e., based on the result of the above indices) was also developed. In addition to this, the results were compared by classifying the risk of slope failure using five classes: unstable, low stability (dangerous), moderate stability, good stability, and safe. The outcomes of this paper are as follows:

- Application of multiple linear regression (MLR), multi-layer perceptron (MLP), radial basis function regression (RBFR), improved support vector machine using sequential minimal optimization algorithm (SMO-SVM), lazy k-nearest neighbor (IBK), random forest (RF), and random tree (RT) were viable for appraising the stability of a single-layered cohesive slope.
- A good level of accuracy was achieved by all applied models based on the obtained values of R^2 (0.9586, 0.9937, 0.9948, 0.9529, 1.000, 0.9997, and 0.9998), *RMSE* (28.4887, 1.7366, 0.7131, 0.6231, 1.9183, 0.0000, 0.1486, and 0.1111), and *RRSE* (11.6985%, 10.2221%, 31.4703%, 0.0000%, 2.4385%, and 1.8221%) for the training dataset, and R^2 (0.9649, 0.9939, 0.9955, 0.9653, 0.9837, 0.9985, and 0.9950), *MAE* (1.1939, 0.5149, 0.3936, 1.0360, 0.8184, 0.2152, and 0.3492), and *RAE* (24.1272%, 10.4047%, 7.9549%, 20.9366%, 16.5388%, 4.3498%, and 7.0576%) for the testing dataset obtained for MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT, respectively.
- The models that use ensemble-learning (i.e., RF and RT) outperformed the models that are based on lazy-learning (IBK), regression rules (MLR and RBFR), neural-learning (MLP), and statistical theories (SMO-SVM).
- Referring to the acquired total scores (the summation of scores obtained for the training and testing stages) of 15, 35, 48, 15, 50, 60, and 57 for MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT, respectively, the superiority of the RF algorithm can be seen. After that, RT, IBK, and RBFR showed an excellent performance in estimating the FOS. Also, satisfactory reliability was achieved for the MLP, SMO-SVM, and MLR methods.
- Based on the results obtained for the classification of the risk of failure, the closest approximation was achieved by the RF, RT, IBK, and SMO-SVM methods. In this regard, 14.60% of the dataset was assorted as being hazardous slopes (i.e., “unstable” and “low stability” categories). This value was estimated as 18.25%, 12.54%, 12.70%, 14.76%, 14.60%, 13.81%, and 14.13% by the MLR, MLP, RBFR, SMO-SVM, IBK, RF, and RT models, respectively.

Author Contributions: H.M. and B.K. performed experiments and field data collection; H.M. and L.K.F. wrote the manuscript, L.K.F. discussion and analyzed the data. B.K. and D.T.B. edited, restructured, and professionally optimized the manuscript.

Funding: This work was financially supported by Ton Duc Thang University and University of South-East Norway.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pourghasemi, H.R.; Pradhan, B.; Gokceoglu, C. Application of fuzzy logic and analytical hierarchy process (AHP) to landslide susceptibility mapping at Haraz watershed, Iran. *Nat. Hazards* **2012**, *63*, 965–996. [[CrossRef](#)]
2. Latifi, N.; Rashid, A.S.A.; Siddiqua, S.; Majid, M.Z.A. Strength measurement and textural characteristics of tropical residual soil stabilised with liquid polymer. *Measurement* **2016**, *91*, 46–54. [[CrossRef](#)]
3. Moayedi, H.; Huat, B.B.; Kazemian, S.; Asadi, A. Optimization of tension absorption of geosynthetics through reinforced slope. *Electron. J. Geotechn. Eng. B* **2010**, *15*, 1–12.
4. Marto, A.; Latifi, N.; Janbaz, M.; Kholghifard, M.; Khari, M.; Alimohammadi, P.; Banadaki, A.D. Foundation size Effect on modulus of subgrade Reaction on sandy soils. *Electron. J. Geotech. Eng.* **2012**, *17*, 2523–2530.
5. Gao, W.; Wang, W.; Dimitrov, D.; Wang, Y. Nano properties analysis via fourth multiplicative ABC indicator calculating. *Arabian J. Chem.* **2018**, *11*, 793–801.
6. Gao, W.; Dimitrov, D.; Abdo, H. Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs. *Discrete Continuous Dyn. Syst.* **2018**, *12*, 711–721.
7. Raftari, M.; Kassim, K.A.; Rashid, A.S.A.; Moayedi, H. Settlement of shallow foundations near reinforced slopes. *Electron. J. Geotech. Eng.* **2013**, *18*, 797–808.
8. Nazir, R.; Ghareh, S.; Mosallanezhad, M.; Moayedi, H. The influence of rainfall intensity on soil loss mass from cellular confined slopes. *Measurement* **2016**, *81*, 13–25. [[CrossRef](#)]
9. Javankhoshdel, S.; Bathurst, R.J. Simplified probabilistic slope stability design charts for cohesive and cohesive-frictional (c- ϕ) soils. *Can. Geotech. J.* **2014**, *51*, 1033–1045. [[CrossRef](#)]

10. Duncan, J.M. State of the art: Limit equilibrium and finite-element analysis of slopes. *J. Geotech. Eng.* **1996**, *122*, 577–596. [[CrossRef](#)]
11. Kang, F.; Xu, B.; Li, J.; Zhao, S. Slope stability evaluation using Gaussian processes with various covariance functions. *Appl. Soft Comput.* **2017**, *60*, 387–396. [[CrossRef](#)]
12. Basarir, H.; Kumral, M.; Karpuz, C.; Tutluoglu, L. Geostatistical modeling of spatial variability of SPT data for a borax stockpile site. *Eng. Geol.* **2010**, *114*, 154–163. [[CrossRef](#)]
13. Secci, R.; Foddis, M.L.; Mazzella, A.; Montisci, A.; Uras, G. Artificial Neural Networks and Kriging Method for Slope Geomechanical Characterization. In *Engineering Geology for Society and Territory-Volume 2*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 1357–1361.
14. Moayedi, H.; Nazir, R.; Mosallanezhad, M.; Noor, R.B.M.; Khalilpour, M. Lateral deflection of piles in a multilayer soil medium. Case study: The Terengganu seaside platform. *Measurement* **2018**, *123*, 185–192. [[CrossRef](#)]
15. Gao, W.; Guirao, J.L.; Abdel-Aty, M.; Xi, W. An independent set degree condition for fractional critical deleted graphs. *Discrete Continuous Dyn. Syst.* **2019**, *12*, 877–886.
16. Gao, W.; Guirao, J.L.; Basavanagoud, B.; Wu, J. Partial multi-dividing ontology learning algorithm. *Inf. Sci.* **2018**, *467*, 35–58.
17. Zhang, Y.; Dai, M.; Ju, Z. Preliminary discussion regarding SVM kernel function selection in the twofold rock slope prediction model. *J. Comput. Civ. Eng.* **2015**, *30*, 04015031. [[CrossRef](#)]
18. Moayedi, H.; Tien Bui, D.; Gör, M.; Pradhan, B.; Jaafari, A. The feasibility of three prediction techniques of the artificial neural network, adaptive neuro-fuzzy inference system, and hybrid particle swarm optimization for assessing the safety factor of cohesive slopes. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 9.
19. Gao, W.; Wu, H.; Siddiqui, M.K.; Baig, A.Q. Study of biological networks using graph theory. *Saudi J. Biol. Sci.* **2018**, *25*, 1212–1219.
20. Moayedi, H.; Hayati, S. Modelling and optimization of ultimate bearing capacity of strip footing near a slope by soft computing methods. *Appl. Soft Comput.* **2018**, *66*, 208–219. [[CrossRef](#)]
21. Bui, D.T.; Moayedi, H.; Gör, M.; Jaafari, A.; Foong, L.K. Predicting slope stability failure through machine learning paradigms. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 395.
22. Jellali, B.; Frikha, W. Constrained particle swarm optimization algorithm applied to slope stability. *Int. J. Geomech.* **2017**, *17*, 06017022. [[CrossRef](#)]
23. Pei, H.; Zhang, S.; Borana, L.; Zhao, Y.; Yin, J. Slope Stability Analysis Based on Real-time Displacement Measurements. *Measurement* **2019**, *131*, 686–693. [[CrossRef](#)]
24. Wang, L.; Li, H.; Chi, Y. Prediction of Surface Grinding Quality Based on SMO-SVM. *Electron. Sci. Technol.* **2017**, *4*, 29.
25. Enyinnaya, R. Predicting Cancer-Related Proteins in Protein-Protein Interaction Networks using Network Approach and SMO-SVM Algorithm. *Int. J. Comput. Appl.* **2015**, *115*, 5–9. [[CrossRef](#)]
26. Long, N.; Gianola, D.; Rosa, G.J.M.; Weigel, K.A.; Kranis, A.; Gonzalez-Recio, O. Radial basis function regression methods for predicting quantitative traits using SNP markers. *Genet. Res.* **2010**, *92*, 209–225. [[CrossRef](#)] [[PubMed](#)]
27. Hervás-Martínez, C.; Gutiérrez, P.A.; Peñá-Barragán, J.M.; Jurado-Expósito, M.; López-Granados, F. A logistic radial basis function regression method for discrimination of cover crops in olive orchards. *Expert Syst. Appl.* **2010**, *37*, 8432–8444. [[CrossRef](#)]
28. Panigrahi, P.K. A Comparative Study of Supervised Machine Learning Techniques for Spam E-Mail Filtering. In Proceedings of the Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, India, 3–5 November 2012; pp. 506–512.
29. Michel, J.; Malik, J.; Inglada, J. Lazy yet efficient land-cover map generation for HR optical images. In Proceedings of the 2010 IEEE International Geoscience and Remote Sensing Symposium, Honolulu, HI, USA, 25–30 July 2010; pp. 1863–1866.
30. Bui, D.T.; Tuan, T.A.; Klempe, H.; Pradhan, B.; Revhaug, I. Spatial prediction models for shallow landslide hazards: A comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree. *Landslides* **2016**, *13*, 361–378.
31. Yilmaz, I. Comparison of landslide susceptibility mapping methodologies for Koyulhisar, Turkey: Conditional probability, logistic regression, artificial neural networks, and support vector machine. *Environ. Earth Sci.* **2010**, *61*, 821–836. [[CrossRef](#)]

32. Krabbenhoft, K.; Lyamin, A.; Krabbenhoft, J. Optum Computational Engineering (Optum G2). Available online: <https://www.optumce.com> (accessed on 1 September 2015).
33. Bui, X.-N.; Muazu, M.A.; Nguyen, H. Optimizing Levenberg–Marquardt backpropagation technique in predicting factor of safety of slopes after two-dimensional OptumG2 analysis. *Eng. Comput.* **2019**, *36*, 1–12. [[CrossRef](#)]
34. Kalantar, B.; Al-Najjar, H.A.; Pradhan, B.; Saeidi, V.; Halin, A.A.; Ueda, N.; Naghibi, S.A. Optimized Conditioning Factors Using Machine Learning Techniques for Groundwater Potential Mapping. *Water* **2019**, *11*, 1909. [[CrossRef](#)]
35. Schaffer, C. Selecting a classification method by cross-validation. *Mach. Learn.* **1993**, *13*, 135–143. [[CrossRef](#)]
36. Naji, H.; Ali, R.H. Qualitative Analysis of Cost Risk Using WEKA Program. In Proceedings of the International Scientific Conference of Engineering Sciences—3rd Scientific Conference of Engineering Science, Diyala, Iraq, 10–11 January 2018; pp. 271–274.
37. Nagar, A. A Comparative Study of Data Mining Algorithms for Decision Tree Approaches using WEKA Tool. *Adv. Nat. Appl. Sci.* **2017**, *11*, 230–243.
38. Seyedashraf, O.; Mehrabi, M.; Akhtari, A.A. Novel approach for dam break flow modeling using computational intelligence. *J. Hydrol.* **2018**, *559*, 1028–1038. [[CrossRef](#)]
39. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [[CrossRef](#)]
40. Makridakis, S.; Wheelwright, S.C.; Hyndman, R.J. *Forecasting Methods and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
41. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [[CrossRef](#)]
42. ASCE Task Committee. Artificial neural networks in hydrology. II: Hydrologic applications. *J. Hydrol. Eng.* **2000**, *5*, 124–137. [[CrossRef](#)]
43. Moayedi, H.; Hayati, S. Artificial intelligence design charts for predicting friction capacity of driven pile in clay. *Neural Comput. Appl.* **2018**, 1–17. [[CrossRef](#)]
44. Moayedi, H.; Rezaei, A. An artificial neural network approach for under-reamed piles subjected to uplift forces in dry sand. *Neural Comput. Appl.* **2017**, *31*, 327–336. [[CrossRef](#)]
45. Broomhead, D.S.; Lowe, D. *Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks*; Royal Signals and Radar Establishment Malvern (United Kingdom): Malvern, UK, 1988.
46. Schaback, R. Multivariate interpolation by polynomials and radial basis functions. *Constr. Approx.* **2005**, *21*, 293–317. [[CrossRef](#)]
47. Powell, M.J.D. Radial basis functions for multivariable interpolation: A review. *Algorithms Approx.* **1987**, *9*, 143–167.
48. Poggio, T.; Girosi, F. Regularization algorithms for learning that are equivalent to multilayer networks. *Science* **1990**, *247*, 978–982. [[CrossRef](#)] [[PubMed](#)]
49. Jayasumana, S.; Hartley, R.; Salzmann, M.; Li, H.; Harandi, M. Kernel methods on Riemannian manifolds with Gaussian RBF kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 2464–2477. [[CrossRef](#)] [[PubMed](#)]
50. Rashidinia, J.; Fasshauer, G.E.; Khasi, M. A stable method for the evaluation of Gaussian radial basis function solutions of interpolation and collocation problems. *Comput. Math. Appl.* **2016**, *72*, 178–193. [[CrossRef](#)]
51. Soleymani, F.; Ullah, M.Z. A multiquadric RBF–FD scheme for simulating the financial HHW equation utilizing exponential integrator. *Calcolo* **2018**, *55*, 51. [[CrossRef](#)]
52. Boztosun, I.; Charafi, A.; Zerroukat, M.; Djidjeli, K. Thin-plate spline radial basis function scheme for advection–diffusion problems. *Electron. J. Bound. Elem.* **2002**, *2*, 267–282.
53. Xiang, S.; Shi, H.; Wang, K.M.; Ai, Y.T.; Sha, Y.D. Thin plate spline radial basis functions for vibration analysis of clamped laminated composite plates. *Eur. J. Mech. A/Solids* **2010**, *29*, 844–850. [[CrossRef](#)]
54. Kavzoglu, T.; Colkesen, I. A kernel functions analysis for support vector machines for land cover classification. *Int. J. Appl. Earth Obs. Geoinf.* **2009**, *11*, 352–359. [[CrossRef](#)]
55. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259. [[CrossRef](#)]
56. Kavzoglu, T.; Sahin, E.K.; Colkesen, I. Landslide susceptibility mapping using GIS-based multi-criteria decision analysis, support vector machines, and logistic regression. *Landslides* **2014**, *11*, 425–439. [[CrossRef](#)]
57. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 2013. [[CrossRef](#)]

58. Cherkassky, V.S.; Mulier, F. *Learning from Data: Concepts, Theory, and Methods*; John Wiley and Sons: Hoboken, NJ, USA, 2006.
59. Patil, S.; Soma, S.; Nandyal, S. Identification of growth rate of plant based on leaf features using digital image processing techniques. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 266–275.
60. Platt, J. Sequential minimal optimization: A fast algorithm for training support vector machines. *CiteSeerX* **1998**, *10*, 4376.
61. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.
62. Raviya, K.H.; Gajjar, B. Performance Evaluation of different data mining classification algorithm using WEKA. *Indian J. Res.* **2013**, *2*, 19–21. [[CrossRef](#)]
63. Bin Othman, M.F.; Yau, T.M.S. Comparison of Different Classification Techniques Using WEKA for Breast Cancer. In Proceedings of the 3rd Kuala Lumpur International Conference on Biomedical Engineering, Kuala Lumpur, Biomed 2006, Kuala Lumpur, Malaysia, 11–14 December 2006. [[CrossRef](#)]
64. Ghosh, S.; Roy, S.; Bandyopadhyay, S. A tutorial review on Text Mining Algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **2012**, *1*, 7.
65. Vijayarani, S.; Sudha, S. Comparative analysis of classification function techniques for heart disease prediction. *Int. J. Innov. Res. Comput. Commun. Eng.* **2013**, *1*, 735–741.
66. Aha, D.W.; Kibler, D.; Albert, M.K. Instance-based learning algorithms. *Mach. Learn.* **1991**, *6*, 37–66. [[CrossRef](#)]
67. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
68. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; CRC Press: Boca Raton, FL, USA, 2007.
69. Hansen, L.K.; Salamon, P. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 993–1001. [[CrossRef](#)]
70. Breiman, L.; Cutler, A. *Random Forests Machine Learning*; Kluwer Academic Publishers: Norwell, MA, USA, 2001; Volume 45, pp. 5–32. [[CrossRef](#)]
71. Pfahringer, B. *Random Model Trees: An Effective and Scalable Regression Method*; Department of Computer Science, The University of Waikato, Private Bag 3105: Hamilton, New Zealand, 2010.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).