**Geomechanics and Engineering**, *Vol. 14, No. 3 (2018) 257-269*
DOI: https://doi.org/10.12989/gae.2018.14.3.257


# Soft computing-based slope stability assessment: A comparative study

A. Kaveh*[1], S.M. Hamze-Ziabari[2] and T. Bakhshpoori[3]

[1]*Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Tehran, Iran*
[2]*Departmant of Civil Engineering, Iran University of Science and Technology, Tehran, Iran*
[3]*Faculty of Technology and Engineering, Department of Civil Engineering, East of Guilan, University of Guilan, Rudsar-Vajargah, Iran*

*(Received October 8, 2016, Revised July 24, 2017, Accepted July 25, 2017)*

**Abstract.** Analysis of slope stability failures, as one of the complex natural hazards, is one of the important research issues in the field of civil engineering. Present paper adopts and investigates four soft computing-based techniques for this problem: Patient Rule-Induction Method (PRIM), M5' algorithm, Group Method of data Handling (GMDH) and Multivariate Adaptive Regression Splines (MARS). A comprehensive database consisting of 168 case histories is used to calibrate and test the developed models. Six predictive variables including slope height, slope angle, bulk density, cohesion, angle of internal friction, and pore water pressure ratio were considered to generate new models. The results of test studies are used for feasibility, effectiveness and practicality comparison of techniques with each other, and with the other available well-known methods in the literature. Results show that all methods not only are feasible but also result in better performance than previously developed soft computing based predictive models and tools. It is shown that M5' and PRIM algorithms are the most effective and practical prediction models.

**Keywords:** slope stability assessment; data mining; PRIM; M5'; GMDH, MARS

## 1. Introduction

The stability of natural and manmade slopes is one of the important research issues because of its disastrous social, economic and environmental consequences such as damages to structures and infrastructures and losses of human lives. A wide range of disciplines from engineering geology to hydrogeology have had a primary focus on this problem. In recent decades this problem is treated as a geotechnical engineering problem with many sources of uncertainties in geotechnical engineering activities such as mining, excavation and transportation construction as well as in all types of trenches, retaining walls and embankments (stockpiles, tailing dams and waste dumps) (Fleurisson and Cojean 2014). A geotechnical engineer may use frequently slope stability analysis via different methods ranging from the Fellenius' method known as the conventional method to upper bound methods. The methods for dealing with the problem mainly include (Cheng and Hoang 2015): analytical methods, numerical methods, expert evaluation, and soft computing-based techniques with some advantages and disadvantages relative to each other. It should be noted that expert evaluation is always needed and included in all the methods. Soft computing based techniques have emerged as more flexible, less assumption dependent and potentially self-adaptive approaches to generate predictive models for problems which by their nature are complex, nonlinear and dynamic (See and Openshaw 1999).

The use of soft computing in the slope stability assessment is a commonly adopted research area and attracted more attention in the recent decade. Sinha and Sengupta firstly demonstrate the applicability of an expert system in slope stability analysis in 1989 (Sinha and Sengupta 1989). Artificial Neural Network was firstly used by Feng *et al.* in 1995 (Xiating *et al.* 1995) and extensively developed for this problem by many others (Lu and Rosenbaum 2003, Wang *et al.* 2005, Cho 2009, Lee *et al.* 2009, Lin *et al.* 2009, Das *et al.* 2011, Abdalla *et al.* 2015). Relevance Vector Machine (RVM) is applied for this problem successfully (Samui 2011, Samui *et al.* 2011, Zhao *et al.* 2012, Zhang *et al.* 2014). Support Vector Machine (SVM) is also utilized successfully for slope stability assessment (Samui 2008, Samui and Kothari 2011, Cheng *et al.* 2012, Cheng and Hoang 2015). The aim of this paper is to assess the potential enhancements to the current knowledge of slope stability assessment that can be achieved through utilizing four relatively new soft computing-based techniques not benefited so far in this problem.

Soft computing-based techniques of choice are: M5' algorithm, Group Method of Data Handling (GMDH), Multivariate Adaptive Regression Splines (MARS), and Patient Rule-Induction Method (PRIM). The M5′ algorithm was originally introduced by Quinlan (Quinlan 1992). One of the advantages of the M5′ compared to other soft computing methods such as ANN or SVM, which are usually vague, is that its predictive model presents rules that are simple and meaningful (Kaveh *et al.* 2016). Among soft computing approaches, the GMDH network and MARS algorithms are known as self-organized and non-parametric

∗Corresponding author, Professor
E-mail: alikaveh@iust.ac.ir

**Copyright © 2018 Techno-Press, Ltd.**

ISSN: 2005-307X (Print), 2092-6219 (Online)

methods to model and discover the behaviors of unknown or complicated systems based on given input–output data points (Ivakhnenko 1971, Friedman 1991, Ivakhnenko and Ivakhnenko 2000). The GMDH and MARS methods also similar to the M5′ algorithm in comparison with ANN method present the dependencies between input and output parameters in a parametric form as an equation while these dependencies are hidden within neural network structures in ANN method. PRIM is a relatively new non-parametric method introduced by Friedman and Fisher (Friedman and Fisher 1999). It has been successfully used in various areas such as geology, marketing, management, finance, medicine, bioinformatics and process optimization. The main features of the PRIM algorithm can be listed as: working with high dimensional data, being parsimonious with data, handling missing values in a non-ad hoc manner, and being based on solid statistical ideas (Nannings *et al.* 2008). PRIM can be more practical in those engineering problems in which finding subgroups with interesting region in high-dimensional data is vital.

Soft computing-based assessment of slope stability analyses like other problems results in predictive models using a set of data samples available from past events. The developed models should be relatively stable and reflect the real variety law of input parameters on output variables (Li and Wang 2010). Input parameters are the influential parameters mainly affect the slope failure. The output of models can be the actual state of the slopes: stable and unstable. Many of the current studies, as referred in the second paragraph, suffer from low quality and small size of samples. Very recently Hoang and Pham (2016) combined firefly algorithm and the least squares support vector classification to establish an integrated slope prediction model considering 6 input influential parameter. They have collected a comprehensive set of 168 data samples available from previous research works. They have also developed three variants of ANN and three other benchmark methods to better verify their proposed hybrid method. It is reported that their proposed hybrid model results in the best prediction performance.

In the artificial intelligence literature, it is accepted that with the accumulation of practical engineering data, the variety law of input parameters on output variables predicted by these techniques becomes increasingly accurate (Li and Wang 2010). It seems that compiling a comprehensive database is requisite in generating new predictive models. Although the main objective of this study is assessing the feasibility, effectiveness and practicality comparison of the four new techniques, but the same database used by Hoang and Pham (Hoang and Pham 2016) is benefited here to more accurate evaluation of the developed models. Using the same dataset (with high quality and size of samples) enables us to make a fair comparison with four different methods developed by them.

The remaining of the paper is organized as follows. Next section outlines the algorithms based on the considered methods after introducing the data base and selected statistical analysis parameters. Section 3 develops models and briefly discusses the results independently for each model. The forth section presents results and discussions for further evaluation the methods and their comparison. The penult section applies and evaluates the developed models for an engineering application: The Yodonghe landslide. At the end, the paper is concluded in Section 6.

## 2. Materials and methods

In the present study, the M5′, MARS, GMDH, and PRIM algorithms are used to predict the stability of slopes. Details of the selected algorithms are presented in the following subsections:

### *2.1 M5′ model tree*

M5′ algorithm is an efficient technique for analyzing complex systems with very high dimensionality-up to hundreds of attributes. Quinlan (Quinlan 1992) presented the M5 algorithm to solve regression and classification problems. Later, Wang and Witten (Wang and Witten 1996) improved the M5 algorithm to so-called M5′ algorithm. The M5′ algorithm divides a complex problem into a number of simple sub-problems and provides the response to a combination of the solutions of these sub-problems. The M5′ algorithm generally includes three processes: (i) building the initial tree (ii) pruning the tree (iii) smoothing. The initial tree is constructed by dividing data space into smaller subspaces based on divide and conquer method (Bhattacharya and Solomatine 2005).

There are some splitting values that divide the entire data sets to several subsets. These splitting values are selected from input variables that maximize the expected error reduction at each node. The standard deviation reduction, SDR, is calculated as a measure of the error at each node as follows (Kaveh *et al.* 2017)

$$SDR = sd\,(\mathrm{T}) - \sum_i \frac{|T_i|}{|T|} \times sd\,(\mathrm{T}_i) \tag{1}$$

where $T$ is the set of records that reach the node, $T_i$ is the resulted set from splitting the node according to the selected attribute, and *sd* is the standard variation. The splitting procedure ceases when the class values of all instances that reach a node vary by less than 5% of the standard deviation of the original instance set, or when only a few instances remain. After building the tree, a multivariate linear regression is created at the bottommost subspace. An over-fitting problem may occur during constructing the model tree and increasing its prediction accuracy for training set. In order to avoid or reduce this problem, pruning procedure can be used. This procedure uses an estimate of the expected error that will be experienced at each node for the test data. First, the absolute difference between the predicted value and the actual output value is averaged for each of the training instances that reached that node. Since the tree has been built exclusively for this dataset, the average value might underestimate the expected error for a new dataset. To compensate this problem, the output value is multiplied by the factor $(n+v)/(n-v)$, where $n$ is the number of training instances that reach the node and $v$ is the number of attributes in the model that represent the output value at that node. If the estimated error at a leaf is higher than at the parent, the leaf node can be dropped (Witten and

Frank 2005).

At final step, smoothing process is applied to reduce the problem of sharp discontinuousness at the leaves of the pruned tree. In the smoothing process, estimated value of each leaf is filtered along the path back to the root. The value at each node that is joining with the predicted value of the linear model for that node is calculated as follows

$$P' = \frac{np + kq}{n + k} \quad (2)$$

where $P'$ is the prediction passed up to the next higher node, $p$ is the prediction passed to the current node from the below, $q$ is the value predicted by model at the node, $n$ is the number of training instances reach to previous node, and $k$ is known as Wang and Witten constant. Finally, the M5′ algorithm yields a set of linear multivariable equations (rules) to estimate the target value (Kaveh *et al.* 2017).

### 2.2 MARS

MARS is a robust nonlinear and nonparametric data mining approach proposed by Friedman in 1991 (Friedman 1991). The nonlinear relationship between the inputs and outputs of a system can be modeled by using a series of piecewise linear or cubic segments (splines). Each linear or cubic segment is presented by an equation known as the basis functions (BFs). The slope of regression functions in linear piecewise model or the concavity of each curve in cubic splines can be varied from one segment to the next. The end points of each segment are called knots. A knot marks the end of one region of data and the beginning of another. Unlike the widely-used parametric linear regression analysis, the MARS algorithm provides more flexibility in exploring the nonlinear relationship between input variables and an output variable. It detects the nonlinear relationship between input and output variables without requiring additional effort to verify a priori assumption about their relationship, unlike the conventional regression approach. To achieve this, MARS searches the possible interactions between variables by checking all degrees of interactions. As a result, all functional forms and interactions between involved variable can be allowed to consider by the MARS algorithm. This feature of the MARS is more critical for the problems in which tracking the complex data structures is needed for high-dimensional datasets. The general MARS function can be expressed using the following equation (Kaveh *et al.* 2017)

$$\tilde{f}(x) = \beta_0 + \sum_{m=1}^{M} \beta_m \lambda_m(x) \quad (3)$$

where $\tilde{f}(x)$ is the predicted response, $\beta_0$ and $\beta_m$ are parameters which are estimated to yield the best data fit and $M$ is the number of BFs included into the model. The basis function in MARS model can be either one single spline function or a product of two or more spline functions for different predictor variables. The spline basis function, $\lambda_m(x)$, can be specified as

$$\lambda_m(x) = \prod_{k=1}^{k_m} [s_{km}(x_{v(k,m)} - t_{k,m})] \quad (4)$$

where $k_m$ is the number of knots, $s_{km}$ takes either 1 or -1 and indicates the right/left regions of the associated step function, $v(k,m)$ is the label of the predictor variable and $t_{k,m}$ is the location of the knot.

To generate basis functions, the MARS applies a searching algorithm in a stepwise manner. The knot locations are selected based on an adaptive regression algorithm. An optimal MARS is developed through a two-stage forward and backward procedure. In the forward stage, MARS over-fits data by considering a great number of BFs. In the backward, to avoid the over-fitting problem, redundant basis functions are deleted from Eq. (3). MARS adopts Generalized Cross-Validation (GCV) to delete the redundant basis functions. The expression of GCV is given as

$$GCV = \frac{\frac{1}{N}\sum_{i=1}^{N}\left[y_i - \hat{f}(x_i)\right]^2}{\left[1 - \frac{C(B)}{N}\right]^2} \quad (5)$$

in which $N$ is the number of data and $C(B)$ is a complexity penalty that increases with the number of basis function (BF) in the model. It is defined as

$$C(B) = (B+1) + dB \quad (6)$$

where $d$ is a penalty for each BF included into the model and $B$ is the number of BFs.

### 2.3 GMDH

GMDH is a learning machine based on the polynomial theory of complex systems (Ivakhnenko 1971). For this network, the most significant input parameters, the number of layers, the number of neurons used in middle layers, and optimal topology design of the network are defined automatically. Therefore, the GMDH network can be categorized as a self-organized model because of these active neurons. The structure of the GMDH network is determined based on a polynomial model in the training stage. The polynomial model that produces the minimum error between the predicted value and observed output is chosen as the best structure. The formal definition of the system identification problem is to find an approximate function $\hat{f}$ that can be used to predict the actual output $\hat{y}$ or a given input vector consisting of $n$ input variables ($X=(x_1,x_2,...,x_n)$) as close as possible to the actual output. Therefore, $M$ observations of multi-input-single-output data pairs are considered as

$$y_i = f\left(x_{i1}, x_{i2}, x_{i3}, \ldots x_{in}\right) \quad \left(i = 1, 2, \ldots, M\right) \quad (7)$$

The general relationship between input and output variables can be expressed by a complicated discrete form of the Volterra function, a series in the form of

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} w_{ijk} x_i x_j x_k + \ldots \quad (8)$$

which is known as the Kolmogorov-Gabor polynomial (Amanifard *et al.* 2008). In the present study, a quadratic polynomial of the GMDH network is used that is written as

$$Quadratic:\ \hat{y} = w_0 + w_1 x_i + w_2 x_j + w_3 x_i x_j + w_4 x_i^2 + w_5 x_j^2 \quad (9)$$

The weighting coefficients of Eq. (9) are calculated using regression techniques such that the difference between actual output ($y$) and the calculated value ($\hat{y}$) for each pair of $x_i$ and $x_j$ as input variables is minimized. In this way, the weighting coefficients of the quadratic function, ($G_i$) are obtained to optimally fit the output to the whole set of input-output data pairs, which is defined as

$$E = \frac{\sum_{i=1}^{M}\left[y_i - G_i()\right]^2}{M} \rightarrow \min \quad (10)$$

In this study, the GMDH network is improved using a back propagation algorithm. This method includes two main steps: (1) the weighting coefficients of the quadratic polynomial are determined using the least squares method from the input layer to output layer in the form of a forward path; and (2) the weighting coefficients are updated using a back-propagation algorithm in a backward path. This procedure may be continued until the error of the training network ($E$) is minimized.

### *2.4 PRIM*

Patient rule-induction method (PRIM) is a soft computing based method introduced by Friedman and Fisher (Friedman and Fisher 1999). It is also referred to as a bump-hunting (or subgroup discovery) technique. Bump-hunting algorithms are employed to divide the input variable space (or covariate space) into sub-regions that the highest or lowest mean value for the outcome occurred in them. These sub-regions are described by simple rules; as the sub-regions are unions of rectangles in the input space. The size of observations in each sub-region is determined by a given threshold. A formal setup of PRIM algorithm is as follows. Let $\{x_i, y_i\}$ be a vector of $N$ observations from some joint distribution with unknown probability density p($y$|$x$), where $y$ denotes the output variable and $x$ a vector consisting of p input variables, $x_p=\{x_1, x_2, \ldots, x_p\}$. The variable $y$ can be considered discrete (e.g., $y$=1 if slope is stable and 0, otherwise) or continuous which the discrete case is considered here (Kaveh, Hamze-Ziabari *et al.* 2016).

The conceptual idea behind of the PRIM algorithm is to seek for a region $B$ (called box) in the input space, in which the mean of the output variable is remarkably larger than the population's mean. Box boundaries are determined by a conjunction of conditions on input variables. A p-dimensional box $B$ as the conjunction of sub-regions of the input variables can be defined as

$$B = [l_1, u_1] \times [l_2, u_2] \times \ldots \times [l_p, u_p] \quad (11)$$

where, $[l_j, u_j]$ is a sub-region of input variables $x_j$. The final structure of the PRIM algorithm is stated as a set of boxes with different rules. When the $i$th observation of input variable of the rules of box $B$ is satisfied, the input vector of that observation is denoted by $x_i \in B$. Once a box is constructed, its observations are removed from modeling process and new boxes are generated based on the remaining observations. Each box has two major statistic

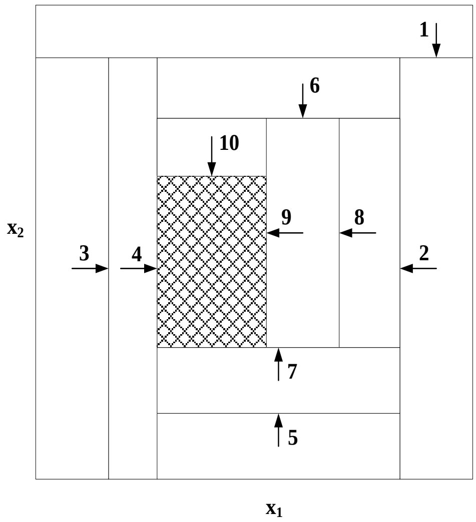



Fig. 1 An illustration of how PRIM finds the first subgroup in 10 steps in a two-dimensional space

properties. The first one is the support for both training and testing datasets, which shows the proportion of observations contained in the box by considering the whole dataset and is calculated as

$$\beta_B = \frac{1}{N}\sum_{i=1}^{N} 1(x_i \in B) \quad (12)$$

where the function 1(.) is equal to 1 when $x \in B$, and zero otherwise. The second property is the box outcome that represents the predicted values in that box. In fact, the algorithm tries to maximize the following objective and attributes the maximum of this objective to the mentioned box.

$$Obj_B = Ave(y_i \mid x_i \in B) \quad (13)$$

where the function *Ave*(.) returns the average of $y$ for the observations that satisfy the conditional argument.

PRIM algorithm finds the optimal boxes by solving the following optimization problem under a constraint on the support of the box $B$ as follows

$$\begin{aligned} &\max_B \quad \mathrm{Obj}_B \\ &\text{subject to } \beta_B \geq \beta \end{aligned} \quad (14)$$

where $\beta$ is a desired minimum value of support. To solve this problem, the PRIM algorithm carries out two main processes to generate a set of boxes: (i) peeling (ii) pasting. PRIM uses an iterative process (called peeling) to create a box by excluding observations with particular values of outcome. In this step, the entire sample space is assumed as an initial box, which can be rectangular in the case of two-dimensional problems and hypercube in general. It then starts to shrink each face of hypercube based on a removing criterion ($\alpha$) that is directly specified by the user. α criterion is defined as a percentage of observations for the variable at that face. It is suggested that α should be selected in the range of 0.05 to 0.1 (Friedman and Fisher 1999). The patient strategy peels the initial box so that the fraction of observation excluded from the reduced box is less than α parameter at each peeling step. For further illustration, the steps taken by PRIM to discover a sub-region with a high density of output variable are shown in Fig. 1.

After the peeling stage, pasting stage uses an iterative procedure to amend the observations for the generated box based upon values of predictor variables. All possible ranges of predictor variables for finding the optimum output, which is typically defined as the largest cumulative incidence, y, are considered at each step of the peeling and pasting stages. The final result of peeling and pasting procedures are presented as a sequence of boxes, including all the boxes involved in the modeling process: from the initial box containing all the data to the box that is obtained after pasting.

When a box (subgroup) is determined and fixed, the PRIM algorithm removes all the associated observations in that box and searches for a new box based on the remaining data. Subgroups are always established based on those obtained earlier: in other words, users should first remove the data corresponding to the earlier boxes to estimate a mean outcome of a box (sequential manner should be implemented).

## 3. Model development

In this section, the database used is introduced and the predictive variables used to develop new models are discussed. The models and their modeling processes based on the M5′, MARS, GMDH, and the PRIM algorithms are presented and discussed.

### 3.1 Selection of input parameters

Very recently Hoang and Pham developed a metaheuristic-optimized least squares support vector classification for slope stability assessment (Hoang and Pham 2016). They have collected a comprehensive set of 168 data samples available from previous research works (comprised of a number of 46 cases based on (Sah, Sheorey *et al.* 1994), 9 cases based on (Lu and Rosenbaum 2003), 31 cases based on (Zhou and Chen 2009), 54 cases based on (Li and Wang 2010), and 30 cases based on (Xiaoming and Xibing 2011)). They have also developed three variants of ANN and three other benchmark methods to better verify their proposed hybrid method. It is accepted in the artificial intelligence literature that with the accumulation of practical engineering data, the variety law of input parameters on output variables predicted by these techniques becomes increasingly accurate (Li and Wang 2010). It seems that compiling a comprehensive database is requisite in generating new predictive models. Although the main objective of this study is assessing the feasibility, effectiveness and practicality comparison of the four new techniques, however, the same database used by Hoang and Pham (Hoang and Pham 2016) is benefited here to more accurate evaluation of the developed models. Using the same dataset (with high quality and size of samples) enables us to make a fair comparison with four different methods developed by them. Database used contains 168 field case histories of which 84 are from the stable slope and the remaining 84 are from unstable cases. Details of the data used were previously published in (Hoang and Pham 2016). The initial step in developing a model was defining the

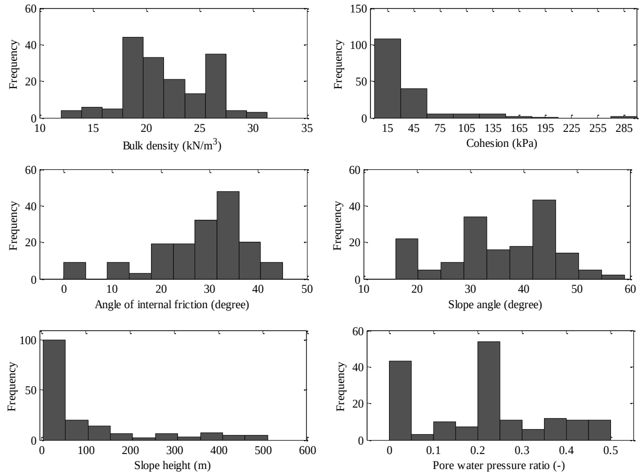



Fig. 2 The histograms of the input variables

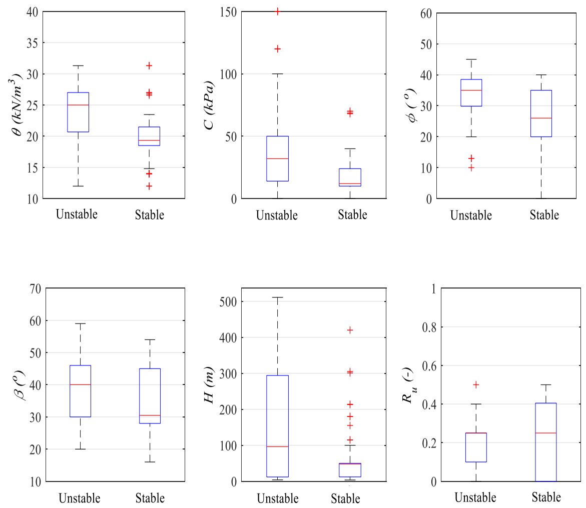



Fig. 3 Boxplots of various predictive variables for stable and unstable slopes based on the whole dataset

Table 1 The ranges of predictive variables for training and testing datasets

| Parameter | Dataset | Min | Max | Mean | Std. |
|---|---|---|---|---|---|
| $\theta$ (kN/m$^3$) | Training | 12 | 31.3 | 21.45 | 3.81 |
| | Testing | 12 | 31.3 | 22.35 | 4.72 |
| $C$ (kPa) | Training | 0 | 300 | 34.17 | 49.17 |
| | Testing | 0 | 200 | 34.02 | 39.17 |
| $\Phi$ (°) | Training | 0 | 45 | 27.96 | 11.30 |
| | Testing | 0 | 45 | 30.23 | 8.96 |
| $\beta$ (°) | Training | 16 | 59 | 35.37 | 10.51 |
| | Testing | 16 | 58 | 37.56 | 9.60 |
| $H$ (m) | Training | 3.6 | 511 | 98.13 | 134.43 |
| | Testing | 3.66 | 511 | 116.29 | 130.67 |
| $R_u$ (-) | Training | 0 | 0.5 | 0.21 | 0.16 |
| | Testing | 0 | 0.5 | 0.22 | 0.15 |

predictive variables that were expected significantly influence on the stability status of slopes. Six input parameters including bulk density ($\theta$), cohesion ($C$), angle of internal friction ($\emptyset$), slope angle ($\beta$), slope height ($H$), pore water pressure ratio ($R_u$) were chosen as main parameters in slope stability analysis based on previous studies (Bishop and Morgenstern 1960, Erzin and Cetin 2014) which are used also by Hoang and Pham (Hoang and Pham 2016) . The histograms of input variables are illustrated in Fig. 2. As shown, the data used contains a wide range of predictive variables. It should be noted that the results of the present study are more reliable in the ranges in which data points are more concentrated according to Fig. 2. The output variable is also selected as SI=1 for stable slopes and SI=0 for unstable slopes.

Fig. 3 uses box plots to compare the different predictive variables for the stable and unstable slopes. To generate each box, the predictive variables for the entire dataset are sorted from the largest to smallest values and then the median value of the sorted dataset is calculated. This median value specifies the central mark (red line) of the box. The edges of the box are 25$^{th}$ and 75$^{th}$ percentiles. The lines extending above and below of each box are defined as the whiskers. The whiskers are drawn from the ends of the interquartile ranges to the furthest observations within the whisker length (the adjacent values). Observations beyond the whisker length are marked as outliers. An outlier is a value that is more than 1.5 times the interquartile range away from the top or bottom of the box. Outliers are shown with a red + sign in Fig. 3. Based on this figure, the variables that exhibit the largest difference between stable and unstable slopes are internal friction angle, slope height, cohesion, and slope angle. The differences between ranges of internal friction angle for stable and unstable cases are remarkable. As shown, the ranges of $\emptyset$ for stable cases are more than unstable cases, indicating that this parameter has a positive effect on stability and increases the resisting forces acting on a slope. The same behavior can also be observed for cohesion parameter. Based on previous studies, it is expected that $\theta$, $\beta$, $H$, and $R_u$ have negative effects on the stability of slopes while the box plots of these parameters show different behavior for the recorded case histories (Florkiewicz and Kubzdela 2013). This contradiction can be attributed to the fact that these parameters can have interaction with each other. For example, very recently based on statistical technique of experimental design, it is stated that the effect of cohesion is highly dependent on $\beta$, $H$, $\emptyset$ and $\theta$ (Kostić, Vasović *et al.* 2016). It should also be noted that the difference between box plots of stable and unstable slopes for pore pressure ratio is negligible.

To develop new models, the whole dataset is randomly divided into two parts: training and testing datasets. Out of 168 historical cases reported in different sources in literature, 2/3 of recorded cases (112 data points) are used to train machine learning algorithms. The remaining 56 records are applied to verify the predictive capability of developed models. The ranges of input variables for training and testing datasets are presented in Table 1. This table contains the minimum (Min), maximum (Max), average (Mean), and standard deviation (Std) of each predictive variables.

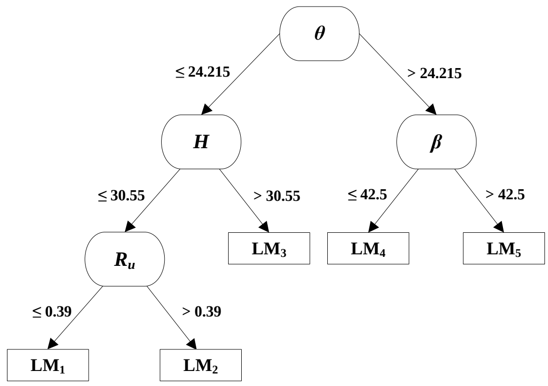
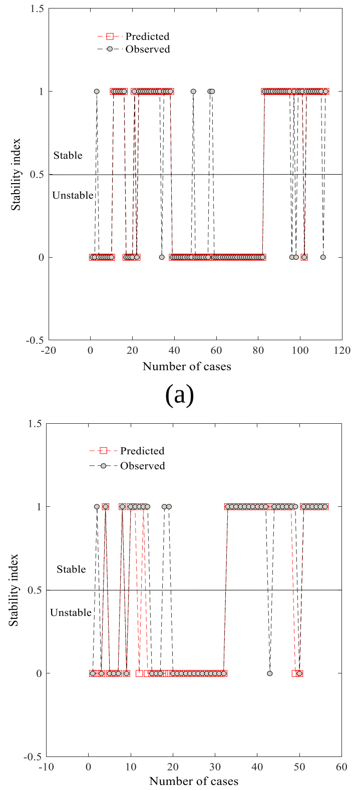



Fig. 4 The developed tree based on the M5′ algorithm




(a)

(b)

Fig. 5 Results of the M5′ model for (a) Training dataset and (b) Testing dataset

To evaluate the performance of the proposed models, the following criteria are employed

$$OA = \frac{(TP + TN)}{(\mathrm{TP + TN + FP + FN})} \tag{15}$$

$$P = \frac{(TP)}{(\mathrm{TP + FP})} \tag{16}$$

$$R = \frac{(TP)}{(\mathrm{TP + FN})} \tag{17}$$

$$F_\beta = \frac{(1+\beta^2)(P \times R)}{(\beta^2 P + R)} \tag{18}$$

where OA=overall accuracy; P=precision; R=recall; TP=total number of stable cases that have been correctly predicted; TN=total number of unstable cases that have been correctly predicted; FP=sum of unstable instances that classified as stable; FN=sum of stable instances that is classified as unstable; and $\beta$ is a measure of the importance of recall to precision and can be defined by the user for a specific project.

### 3.2 Developed M5′

Following the data division, the training data points were presented to the M5′ algorithm for model development. The main concept behind the M5′ algorithm is to divide a complex problem into several subspaces in which the response variables can easily be estimated. One of the main parameters that should be determined by users in M5′ algorithm is the minimum number of instances in each subspace. The recommended value for this parameter is 4.0 by (Quinlan 1992). However, this parameter should be chosen based on the performance of the developed model for testing dataset. Therefore, this should be done to ensure that the over-fitting problem would not occur in the modeling process. For slope stability problem, this value was chosen as 15 based on the results of different developed models. According to this value, there are at least 15 historical cases in each developed class. The structure of the developed model tree is shown in Fig. 4. The M5′ algorithm only presents a linear relationship between input and output variables in each class. The developed multivariate linear regressions in each class are presented in Eqs. 19 (a)-(e).

As shown, the M5′ yields transparent and compact formulas like empirical equations for prediction of the slope stability status while other soft computing methods such as ANN act like a black box. These equations also have good interpretability. For example, the stability of slope increases when internal friction angle increases or slope angle decreases in all relations. Furthermore, the unit weight of soil can have positive or negative effects on slope stability based on Eqs. 19 (a)-19(e). According to these equations, the unit weight of soil improved the slope stability up to a point (24.2 kN/m$^3$) and then, it worsened the stability status. For large unit weight (>24.2 kN/m$^3$), the slope angle parameter determined the behavior of stability status. It should be noted that these splitting values do not necessarily have any physical interpretation and are obtained by minimizing the prediction error. However, most of the results obtained by the proposed model are in line with the results of the experimental observations and are also physically interpretable.

$$LM_1: \quad \mathrm{SI} = \mathrm{Round}\,(0.0082\theta + 0.0014C + 0.0254\varphi - 0.0319\beta - 0.0009H - 0.1709R_u + 0.7462) \tag{19(a)}$$

$$LM_2: \quad \mathrm{SI} = \mathrm{Round}\,(0.0082\theta + 0.0014C + 0.0181\varphi - 0.0196\beta - 0.0009H - 0.3458R_u + 0.4116) \tag{19(b)}$$

$$LM_3: \quad \mathrm{SI} = \mathrm{Round}\,(0.0082\theta + 0.0019C + 0.0082\varphi - 0.0094\beta - 0.0012H - 0.0034R_u + 0.1103) \tag{19(c)}$$

$$LM_4: \quad \mathrm{SI} = \mathrm{Round}\,(-0.0277\theta + 0.005\varphi - 0.0096\beta + 1.862) \tag{19(d)}$$

$$LM_5: \quad \mathrm{SI} = \mathrm{Round}\,(-0.0789\theta + 0.005\varphi - 0.0089\beta + 3.1028) \tag{19(e)}$$

Results of using the above equations for training and testing datasets are presented in Fig. 5. As can be seen, there is a good agreement between the observed and predicted slope status by using these equations. Furthermore, the performance of the developed M5′ algorithm is approximately similar for both training and testing datasets. The similar performance indicates that the developed model can be applied for the trained ranges.

### 3.3 Developed MARS

The basis functions (BFs) of the MARS algorithm can be piecewise-cubic or piecewise-linear (Friedman 1991). In the present study, the piecewise-linear model is presented only due to having better performance in comparison to the piecewise-cubic model for slope stability problem. One of the main parameters that should be adjusted for developing the MARS algorithm is the Generalized Cross-Validation (GCV) penalty for each knot. This penalty is usually chosen between 2 and 4. In the present study, the recommended value of 3 suggested by Friedman (Friedman 1991) is used. Applying the MARS algorithm for developing a predictive model, the following relationship is obtained for training dataset

$$\begin{aligned}\mathrm{SI} = \mathrm{Round}(&-0.002 - 0.19BF_1 - 0.0071BF_2 + 0.038BF_3 - 0.0014BF_4 - 0.3BF_5 \\ &+ 0.0072BF_6 + 0.017BF_7 - 0.007BF_8 - 0.0059BF_9 + 0.0018BF_{10} + \\ &0.14BF_{11} + 0.0063BF_{12} - 0.01BF_{13} + 0.011BF_{14} - 0.002BF_{15} \\ &-1.5\times10^{-5}BF_{16})\end{aligned} \tag{20}$$

Table 2 presents the BFs of the developed model. As it can be seen, those of 16 BFs, 13 BFs with interaction terms are incorporated in this model, indicating that the model is not simply additive terms and they play a significantly important role in the developed final model. Therefore, the MARS algorithm captures the complex relationships between the input and output variables without requiring an

Table 2 Basis functions of the developed MARS model

| Basis function | Equation |
|---|---|
| $BF_1$ | max(0, $\theta$-27) |
| $BF_2$ | max(0, 46-$H$) × max(0, Ø-30) |
| $BF_3$ | max(0, 70-$C$) × max(0, $\theta$-27) |
| $BF_4$ | max(0, 46-$H$) × max(0, $\beta$-30) |
| $BF_5$ | max(0, 46-$H$) × max(0, 30-$\beta$) × max(0, $R_u$-0.45) |
| $BF_6$ | max(0, 46-$H$) × max(0, 30-$\beta$) × max(0, 0.45-$R_u$) |
| $BF_7$ | $BF_6$ × max(0, $\theta$ -19) |
| $BF_8$ | $BF_3$ × max(0, Ø -35) |
| $BF_9$ | max(0, 70-$C$) × max(0, $\theta$ -27) × max(0, 35- Ø) × max(0, $H$-290) |
| $BF_{10}$ | max(0, $H$-50) |
| $BF_{11}$ | max(0, Ø -39) |
| $BF_{12}$ | max(0, 50-$H$) × max(0, Ø -27) |
| $BF_{13}$ | max(0, 39- Ø) × max(0, $C$ -30) |
| $BF_{14}$ | max(0, 39- Ø) × max(0, $C$ -25) |
| $BF_{15}$ | max(0, 27- $\theta$) × max(0, 22- Ø) × max(0, $C$ -25) |
| $BF_{16}$ | max(0, 70-$C$) × max(0, 180-$H$) × max(0, $\theta$ -22) × max(0, 40- $\beta$) |

Table 3 ANOVA decomposition of the MARS model

| Function No. | STD | GCV | Variables |
|---|---|---|---|
| 1 | 0.081 | 0.095 | $\theta$ |
| 2 | 0.205 | 0.184 | $\Phi$ |
| 3 | 0.228 | 0.232 | $H$ |
| 4 | 0.324 | 0.342 | $\theta\ C$ |
| 5 | 0.412 | 0.505 | $C\ \Phi$ |
| 6 | 0.351 | 0.427 | $\Phi\ H$ |
| 7 | 0.213 | 0.201 | $\beta\ H$ |
| 8 | 0.352 | 0.366 | $\theta\ C\ \theta$ |
| 9 | 0.124 | 0.108 | $\beta\ H\ R_u$ |
| 10 | 0.103 | 0.104 | $\theta\ C\ \Phi\ H$ |
| 11 | 0.138 | 0.124 | $\theta\ C\ \beta\ H$ |
| 12 | 0.139 | 0.126 | $\theta\ C\ H\ R_u$ |

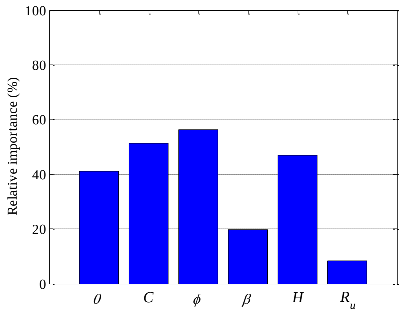
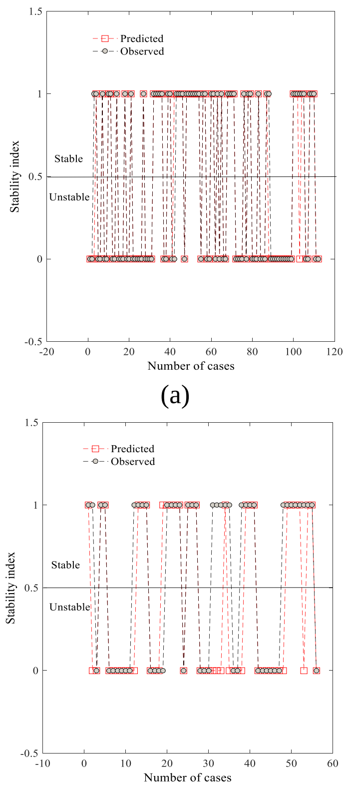



Fig. 6 The relative importance of the input variables in the developed MARS model




(a)

(b)

Fig. 7 Results of the MARS model for (a) Training and (b) Testing datasets

additional effort to verify a priori assumption about the relationship between the set of input variables and output variable. One of the limitations of the M5′ can be attributed to this feature because the algorithm always considers a linear relationship between input and output variables. This feature can be more critical as the dimension and complexity of the problem increase.

Another important and practical advantage of the MARS algorithm is its ability to determine the most influential parameters in developing a process of a predictive model. Table 3 presents the Analysis of Variances (ANOVA) decomposition of the developed MARS models for predicting the stability status of slopes. The first column lists the number of the ANOVA function. The standard deviation of the corresponding ANOVA functions is given in the second column. This gives one indication of the relative contribution of each function to the overall model performance. The third column provides another indication of the importance of the corresponding ANOVA function, by listing the GCV score for a model with the whole basis functions compared to a model in which that particular ANOVA function is removed. This can be used to judge whether this ANOVA function makes an important contribution to the model performance, or whether it only slightly helps to improve the global GCV score. The last column gives the particular predictor variables associated with ANOVA function. This ability of the MARS can be used to specify the most important parameters in slope stability problem. To achieve this, MARS algorithm removes ANOVA functions one by one and then calculates the increase of GCV value caused by removing that ANOVA function. The ANOVA function that increases the GCV value considerably plays important role in developing process of the MARS model. In the same way, variables included in the ANOVA functions also play an important role in developing the process of the MARS model. For example, according to Table 3, the ANOVA functions with parameters of $C$ and $\phi$ have improved the amount of GCV values remarkably. Therefore, it can be expected that these parameters are more important and play more crucial roles than other parameters in developing the MARS model.

Fig. 6 shows the plot of the relative importance of the input variables in developing the MARS model. The angle of internal friction has the most influence and the pore water pressure coefficient has the least influence in the developed MARS model. The cohesion, slope height, bulk density, and slope angle are recognized as the other influential parameters in the order of importance. These results are in line with the field observations based on Fig. 3. The performances of the developed MARS model are demonstrated in Fig. 7. As shown, the developed model has a good predictive ability. However, the accuracy of developed model for testing dataset is less than training dataset.

### 3.4 Developed GMDH

As stated before, the GMDH algorithm presents its predictive model as a quadratic polynomial. It determines the weighting coefficients of a quadratic polynomial using least square method in the forward phase. Then, weighting coefficients are enhanced based on a back propagation

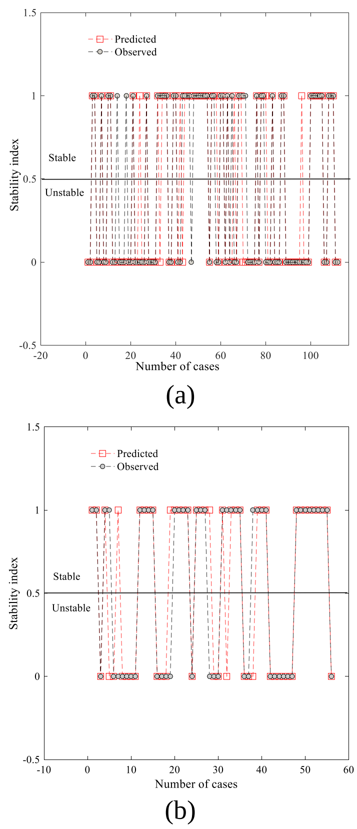



Fig. 8 Results of the GMDH model for (a) Training dataset and (b) Testing dataset

| | |
|---|---|
| Box$_1$:<br>$P_{SI}=1$ Support = 0.059 (10)<br><br>$35 \le \emptyset \le 40$<br>$0.25 \le R_u \le 0.25$<br>$25 \le \theta \le 27$<br>$46 \le C \le 50$<br>$42 \le \beta \le 47$<br>$393 \le H \le 443$ | Box$_2$:<br>$P_{SI}=0.955$ Support = 0.226 (38)<br><br>$\emptyset \ge 13$<br>$10 \le C \le 150$<br>$20.4 \le \theta \le 27.3$<br>$0.15 \le R_u \le 0.35$<br>$\beta \ge 22$ |
| Box$_3$:<br>$P_{SI}=1$ Support = 0.089 (15)<br><br>$\emptyset \ge 20$<br>$18.8 \le \theta \le 28.4$<br>$29.4 \le C \le 150$<br>$R_u \le 0.3$<br>$15 \le H \le 200$<br>$20 \le \beta \le 53$ | Box$_4$:<br>$P_{SI}=1$ Support = 0.059 (10)<br><br>$20 \le \emptyset \le 40$<br>$14.4 \le C \le 57.5$<br>$20 \le \beta \le 40$<br>$10 \le H \le 30.6$<br>$R_u \le 0.38$<br>$18.8 \le \theta \le 20$ |
| Box$_5$:<br>$P_{SI}=1$ Support = 0.041 (8)<br><br>$13 \le \emptyset \le 30$<br>$C \le 35$<br>$20 \le \beta \le 40$<br>$8 \le H \le 12$<br>$18 \le \theta \le 21$<br>$0.3 \le R_u \le 0.5$ | Box$_6$:<br>$P_{SI}=0.75$ Support = 0.059 (9)<br><br>$21.3 \le \emptyset \le 40$<br>$\theta \le 22.4$<br>$30 \le \beta \le 35$<br>$C \le 10.4$<br>$4 \le H \le 37$<br>$R_u \le 0.35$ |
| Box7 'leftovers':<br>$P_{SI}=0.0339$ Support = 0.523 (88) | |

Fig. 9 Box decision developed based on the PRIM algorithm for slope stability problem

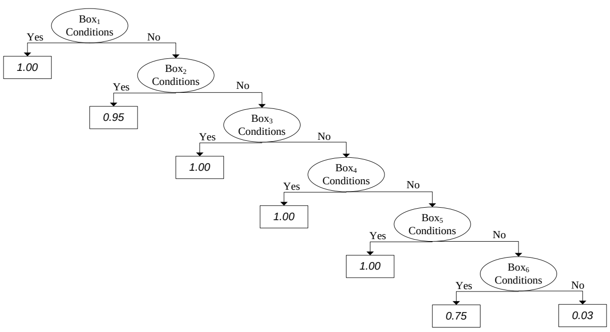



Fig. 10 Illustrative diagram of the predicting procedure based on the developed PRIM model

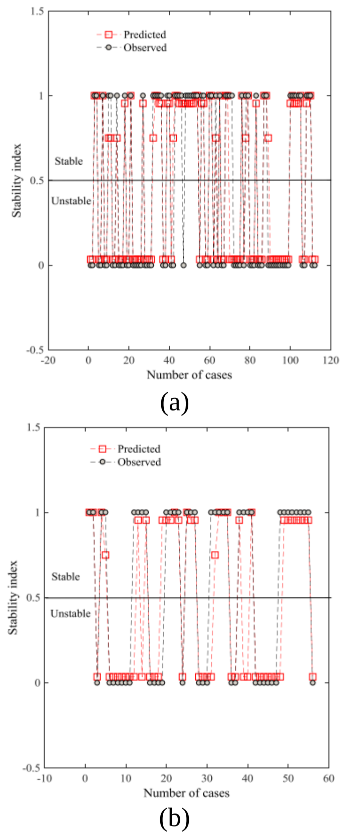



Fig. 11 Results of the PRIM model for (a) Training datset and (b) Testing dataset

algorithm in a backward phase. This procedure could be continued until the error criterion was minimized. In this study, according to training dataset, the GMDH algorithm returned the following selective polynomials for predicting the stability status of slopes

$$\begin{aligned} L_1 = {} & 0.87 - 0.062\beta + 0.097\phi - 0.12\theta - 0.0012\phi\beta + 0.0043\theta\beta \\ & - 0.0047\theta\phi - 0.00016\beta^2 + 0.0013\phi^2 + 0.0035\theta^2 \end{aligned} \tag{21}$$

$$\begin{aligned} \text{SI} = {} & \text{Round}(-0.25 + 1.6L_1 - 0.0037H + 0.015C - 0.0013HL_1 - 0.009CL_1 \\ & + 6.9\times10^{-6}CH - 0.31L_1^2 + 10^{-5}H^2 - 3.4\times10^{-5}C^2) \end{aligned} \tag{22}$$

The criterion for evaluation of neurons and stopping the algorithm was chosen based on the best performance of both testing and training datasets. The algorithm is allowed to generate new layer as long as the error values of developed model get better (smaller). The results of the developed GMDH for training and testing datasets are shown in Fig. 8. From this figure, the predictions of the GMDH had an acceptable success rate for training and testing datasets.

### *3.5 Developed PRIM*

To select an optimum structure for predicting slope stability based on PRIM algorithm, two important parameters known as the peeling quantile ($\alpha$) and the minimum support ($\beta$) should be adjusted. The value of $\alpha$ is recommended between 0.05 and 0.1 (Friedman and Fisher

1999). This parameter defines the fraction of data points in the current box that is removed in each iteration. The larger values of $\alpha$ may lead to increase the risk of missing an optimal box. On the other hand, the smaller values of $\alpha$ can also result in an over-fitting problem in the peeling process. In general, a small value of $\alpha$ makes the algorithm to be more "patient". To find an appropriate value for $\alpha$, the parameter is changed in acceptable ranges between 0.05 and 0.1. The best value for this parameter is selected 0.07 based on the best performances of the developed model for training and testing datasets.

Another important parameter, $\beta$, is a criterion that determines the fraction of data covered by each box. Values that are smaller than 1.0 are treated as the $\beta$ percent of the whole dataset. For example, the value of 0.01 for slope stability problem means that there are at least 0.01×168=16.8 instances in each developed box. In general, the performance of the PRIM algorithm decreases as the $\beta$ parameter increases. The smaller value of $\beta$ may lead to higher complexity of the developed model. To obtain an optimum structure, this parameter is varied between 0.01 and 0.5. The best result obtained for $\beta$=0.05 (i.e. the number of instances in each box should be at least 8 case histories).

By applying $\alpha$=0.07 and $\beta$=0.05, the PRIM algorithm discovers seven subgroups for predicting slope stability status as presented in Fig. 9. Each subgroup consists of three parts: the fraction of total instances assigned to each subgroup (Support), the ranges of predictive variables detected by PRIM for each subgroup, and the probability of being stable ($P_{SI}$). To further illustration, the general procedure of applying the PRIM algorithm to slope stability problem for different ranges of input variables is shown in Fig. 9. Out of 168 records, 10 instances remain in $\text{Box}_1$ during the first peeling and pasting process. The remaining unassigned records (168-10=158) are considered for constructing the second partition. The process of producing a new partition based on unassigned instances continues until all records assigned to a partition. The instances are not included in any high-risk partition are assigned to the remainder partition (unstable cases).

The procedure that decision maker should follow to predict slope stability status based on the developed PRIM algorithm is illustrated in Fig. 10. The PRIM algorithm presents its model in a sequential manner. It means that the historical cases with all given predictive variables are firstly checked for $\text{Box}_1$. This procedure continues until the physical parameters related to the cases satisfy the necessary conditions in one of the boxes. It should be noted that a case can be placed in more than one box but as it is clear from the sequential manner in the Fig. 10, the PRIM predictions are based on the first box that satisfies all its conditions. The first five boxes in the developed PRIM algorithm specify the condition that the probability of being stable is almost 1.00. The $\text{Box}_6$ represents conditions that the probability of being stable is 0.75. The main reason of smaller probability with respect to the other mentioned subgroups can be attributed to the smaller values of cohesion parameter in this box. This parameter plays as resisting factor in slope stability problem. The lower values of cohesion parameter lead to decrease in resisting force and as a result, the PRIM algorithm reduces the safety of the case histories placed in this box by assigning a probability of 0.75. However, the other stabilizing parameters are in safe ranges that are classified this box as a representative of stable slopes. The results of training and testing datasets related to the developed PRIM algorithm are depicted in Fig. 11. As shown, the model performances for both training and testing datasets are remarkably promising.

## 4. Results and discussion

For evaluation of the developed M5′, MARS, GMDH, and PRIM algorithms, as stated in subsection 3.1, the overall accuracy (OA), precision (P), recall (R), and F score statistical parameters related to each model are presented in Table 4. OA parameter is a common metric and gives an overall accuracy of the developed model for both stable and unstable case histories. For example, an accuracy of 0.9 means that 90% of the whole dataset has been correctly estimated. However, it cannot be implied that the developed model has particularly 90% accuracy in prediction of the unstable cases (or stable cases). This deficiency can be more apparent when a class of imbalance exists or the number of instances from each class is not equal in the whole dataset.

To compensate this problem, precision and recall parameters are defined for each class. The efficiency of applying these parameters is significant when there is a remarkable class imbalance in the dataset. *Precision* is an accuracy metric that measures the performance of the model for a single class. A precision of 1.0 for unstable cases indicates that all the predictions of the unstable cases based on the developed model are actually observed as unstable cases in the field while it does not consider unstable cases that are misclassified by the algorithm. On the other hand, *recall* metric measures the accuracy of the developed model only based on the prediction values. A recall of 1.0 means

Table 4 The results of the developed models for training, testing and total datasets

| Model | Data set | OA | Stable | | | Unstable | | |
|---|---|---|---|---|---|---|---|---|
| | | | P | R | F score | P | R | F score |
| GMDH | Training | 0.87 | 0.86 | 0.86 | 0.86 | 0.88 | 0.88 | 0.88 |
| | Testing | 0.89 | 0.90 | 0.90 | 0.90 | 0.87 | 0.87 | 0.87 |
| | Total | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 |
| M5′ | Training | 0.92 | 0.92 | 0.92 | 0.92 | 0.93 | 0.93 | 0.93 |
| | Testing | 0.87 | 0.81 | 0.96 | 0.87 | 0.95 | 0.79 | 0.86 |
| | Total | 0.91 | 0.88 | 0.93 | 0.90 | 0.94 | 0.94 | 0.94 |
| PRIM | Training | 0.95 | 0.96 | 0.94 | 0.95 | 0.95 | 0.96 | 0.95 |
| | Testing | 0.87 | 0.81 | 0.96 | 0.87 | 0.95 | 0.79 | 0.86 |
| | Total | 0.93 | 0.90 | 0.93 | 0.91 | 0.95 | 0.94 | 0.94 |
| MARS | Training | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |
| | Testing | 0.82 | 0.71 | 0.95 | 0.81 | 0.95 | 0.71 | 0.81 |
| | Total | 0.91 | 0.87 | 0.96 | 0.91 | 0.96 | 0.90 | 0.93 |

Table 5 Comparison between the results of 10-fold cross-validation analysis for different soft computing methods

| Model | BDA | SCG-ANN | RMV | SVC | MO-LSVC | M5' | MARS | GMDH | PRIM |
|---|---|---|---|---|---|---|---|---|---|
| | (Hoang and Pham 2016) | | | | | Present study | | | |
| 10-fold cross validation | | | | | | | | | |
| OA | 0.75 | 0.81 | 0.83 | 0.83 | 0.86 | 0.91 | 0.91 | 0.87 | 0.96 |

that the developed model correctly predicts all observed unstable cases while it does not consider the real observed stable cases that misclassified by the model. There is an inverse relationship between recall and precision. An increase in precision may lead to a reduction of recall and vice versa. To combine these statistical parameters, F score parameter is defined as a single statistical evaluation parameter. The higher value of F score means the better performance of the developed model. In the present paper, the harmonic mean of precision and recall using equal weights are employed to measure the performance of the developed model ($\beta$=1).

The ratio between the numbers of stable cases to unstable cases is 0.86:1 for training data set. In fact, there is a class of imbalance problem for training dataset. This ratio for testing and total datasets is 1.33:1 and 1:1, respectively. As mentioned, the OA metric cannot be used alone to measure the performances of the developed models. Therefore, the OA, P, R, and F score parameters are used to evaluate the performances of the M5′, MARS, GMDH and PRIM algorithms. The results of these metric parameters for training, testing and total datasets are presented in Table 4. The OA value for the PRIM algorithm is the highest (0.93) and the GMDH algorithm has the lowest value of OA (0.88) for total case histories. The M5′ and MARS have the same OA values (0.91). It should be noted that all the developed models show a remarkable accuracy in prediction of the slope stability status. However, the performance of the developed PRIM model is slightly better than the other ones according to all statistical metrics presented in Table 4. The prediction of the unstable slope is more critical than a stable one. According to Table 4, the performance of the M5′ and PRIM algorithms are the same based on F score for the whole dataset, and are the highest. The MARS algorithm also shows a promising performance based on F score metric (0.93). The F score value of the GMDH algorithm is the lowest (0.88) for all unstable cases. However, the interesting point about the developed GMDH algorithm is that is its nearly identical performance for training, testing, and total dataset, while the accuracy of the other developed models slightly decreases for testing dataset, especially for the MARS algorithm.

To further evaluation of developed models, their performances are also compared with several soft computing approaches developed by Hoang and Pham (Hoang and Pham 2016). They developed a Metaheuristic-Optimized Least Squares Support Vector Classification (MO-LSVC) by Firefly algorithm to establish an integrated slope prediction model (Hoang and Pham 2016). They have also developed six extra models using other benchmark methods: ANN with three different algorithms such that the model obtained by scaled conjugate gradient algorithm (SCG-ANN) results was better than the two others, a Support Vector Classifier (SVC), a Relevance Vector Machine (RMV), and the Bayes Discriminant Analysis (BDA). Although it is reported that their proposed hybrid model (MO-LSVC) results in the best prediction performance, compared to many prediction performance evaluations. Table 5 compares the overall accuracy (OA) of their best ANN (SCG-ANN), BDA, RMV, SVC and MO-LSVC models with the proposed ones in this study based on 10-fold cross validation technique.

According to Table 5, the MO-LSVC results show a remarkable accuracy based on 10-fold cross validation technique in comparison to BDA, SCG-ANN, RMV, and SVC algorithms. However, all developed models in the present study outperform the mentioned algorithms. The performance of the GMDH and MO-LSVC are nearly the same. However, the M5′, MARS, and PRIM algorithms remarkably outperformed the MO-LSVC algorithm. The performances of the MARS and M5′ algorithms are the same based on 10-fold cross validation technique. In general, the accuracy of the PRIM algorithm is the most superior in this aspect. The main advantage of the developed models, especially the M5′ and PRIM algorithms, is that they present simple and compact relationships between the input and output variables like empirical equations, while other soft computing methods such as BDA, SCG-ANN, RMV, SVC, and MO-LSVC act like black boxes and the user cannot directly reproduce the results without reconstructing the algorithms.

## 5. Engineering application: The Yodonghe landslide

In this section, the proposed models are applied to predict the stability status of the Yodonghe landslide. This landslide was 2300 m away from the Shuibuya dam site and has 18 slopes with high potential to reactivate or fail. The landslide occurred with a volume about 0.0235 km$^3$, which made it as one of the largest prehistoric landslides. The stability evaluation of slopes in the mentioned area was very crucial for damage management and also assessing the geological conditions of dam construction (Wang *et al.* 2005). The upper part of the Yodonghe landslide was one of the most crucial slopes. In this study, this part of landslide is investigated.

The predictive variables including $\theta$, $c$, $\emptyset$, $\beta$, $H$, and $R_u$ were reported as 21.0 kN/m$^3$, 20.0 kPa, 24.0°, 21.0°, 85.0 m, and 0.0 (under the dry condition) along the sliding surface of the upper part of the Yodonghe landslide. After applying these input variables to the M5′ model, the mentioned slope according to the developed tree in the Fig. 4 is classified in $LM_3$, where $\theta \leq 24.21$ and $H$>30.55. By computing the response of the $LM_3$, the result indicates that the M5′ model correctly detects the status of this slope as an unstable slope. The PRIM algorithm also classifies this slope in the last developed box based on its sequential evaluation, which represents an unstable slope. The MARS and GMDH also predict this slope as an unstable slope.

By considering the results of the performance analysis and model development process, the M5′, MARS, GMDH, and PRIM algorithms can be used as optional tools to develop new predictive models in a different domain of geotechnical engineering. The PRIM algorithm can be used successfully in problems with the binary manner or classifying the critical area in a special problem. The M5′ algorithm also can be used as a practical tool to develop a robust and meaningful rule-based predictive model.

## 6. Conclusions

Soft computing approaches such as M5′, MARS, GMDH, and PRIM algorithms are used to develop new predictive models for evaluation of slope stability. Though the factor of safety in slope stability analysis is very important, developing models to predict merely whether a slope is stable or unstable can be very important which is the main objective of this study. A comprehensive database consisting of 168 case histories reported in different studies is used to train the mentioned algorithms. Six basic geometrical parameters and soil factors including slope height, slope angle, bulk density, cohesion, angle of internal friction, and pore water pressure ratio are considered as predictive variables. The performances of developed models are evaluated by using statistical error metrics. The most important outcome of this study can be summarized as follows:

• The M5′ algorithm predicts the stability index based on a rule-based structure. The developed model tree and its equations are simple and meaningful. This feature makes this approach be more practical than most of the soft computing approaches like ANN and SVM. It also predicts the stability index with 91% morality, indicating the effectiveness of approach in the aspect of accuracy.

• The MARS algorithm with 91% morality shows the same performance as the M5′ algorithm. However, the developed MARS model is more complicated than the M5′ algorithm. One of the main advantages of MARS is its ability in finding the most important parameters involved in an unknown problem. The sensitivity analysis is performed based on the MARS algorithm indicating that the angle of internal friction has the most influence and the pore water pressure ratio has the least influence. The cohesion, slope height, bulk density, and slope angle are the other important parameters.

• The overall accuracy of the GMDH algorithm is less than the other developed models with 88% morality. The interesting point about the developed GMDH algorithm is that it has nearly the same performance for training, testing, and total dataset.

• The PRIM algorithm as a new data mining approach in engineering domain is employed to discover the region with high safety from the stability point of view. The result indicates that the PRIM model shows the best performance from the accuracy point of view compared to other developed models.

The accuracy of the developed models is also compared with those of BDA, SCG-ANN, RMV, SVC, and MO-LSVC methods based on 10-fold cross validation technique. The results indicated that all the developed models outperform the mentioned methods in terms of accuracy. In general, it can be concluded that the M5′, MARS, GMDH, and PRIM algorithms can be successfully used as reliable alternative approaches in geotechnical engineering.

## References


Abdalla, J.A., Attom, M.F. and Hawileh, R. (2015), "Prediction of minimum factor of safety against slope failure in clayey soils using artificial neural network", *Environ. Earth Sci.*, **73**(9), 5463-5477.

Aksoy, C.O., Uyar, G.G. and Ozcelik, Y. (2016), "Comparison of Hoek-Brown and Mohr-Coulomb failure criterion for deep open coal mine slope stability", *Struct. Eng. Mech.*, **60**(5),809-828.

Amanifard, N., Nariman-Zadeh, N., Farahani, M.H. and Khalkhali, A. (2008), "Modelling of multiple short-length-scale stall cells in an axial compressor using evolved GMDH neural networks", *Energy Convers. Manage.*, **49**(10), 2588-2594.

Bhattacharya, B. and Solomatine, D.P. (2005), "Neural networks and M5 model trees in modelling water level-discharge relationship", *Neurocomput.*, **63**, 381-396.

Bishop, A.W. and Morgenstern, N.R. (1960), "Stability coefficients for earth slopes", *Geotechnique*, **10**(4), 129-153.

Cheng, M.Y. and Hoang, N.D. (2015), "A swarm-optimized fuzzy instance-based learning approach for predicting slope collapses in mountain road", *Knowl. Based Syst.*, **76**, 256-263.

Cheng, M.Y. and Hoang, N.D. (2015), "Typhoon-induced slope collapse assessment using a novel bee colony optimized support vector classifier", *Nat. Hazards*, **78**(3), 1961-1978.

Cheng, M.Y., Roy, A. F. and Chen, K.L. (2012), "Evolutionary risk preference inference model using fuzzy support vector machine for road slope collapse prediction", *Expert Syst. Appl.*, **39**(2), 1737-1746.

Cho, S.E. (2009), "Probabilistic stability analyses of slopes using the ANN-based response surface", *Comput. Geotech.*, **36**(5), 787-797.

Das, S.K., Biswal, R.K., Sivakugan, N. and Das, B. (2011), "Classification of slopes and prediction of factor of safety using differential evolution neural networks", *Environ. Earth Sci.*, **64**(1), 201-210.

Erzin, Y. and Cetin, T. (2014), "The prediction of the critical factor of safety of homogeneous finite slopes using neural networks and multiple regressions", *Geomech. Eng.*, **6**(1), 1-15.

Feng, X.T., Wang, Y. and Lu, S. (1995), "Neural network estimation of slope stability", *J. Eng. Geol.*, **3**(4), 54-61.

Fleurisson, J.A. and Cojean, R. (2014), *Error Reduction in Slope Stability Assessment*, in *Surface Mining Methods, Technology and Systems*.

Florkiewicz, A. and Kubzdela, A. (2013), "Factor of safety in limit analysis of slopes", *Geomech. Eng.*, **5**(5), 485-497.

Friedman, J.H. (1991), "Multivariate adaptive regression splines", *Annal. Stat.*, **19**(1), 1-67.

Friedman, J.H. and Fisher, N.I. (1999), "Bump hunting in high-dimensional data", *Stat. Comput.*, **9**(2), 123-143.

Hoang, N.D. and Pham, A.D. (2016), "Hybrid artificial intelligence approach based on metaheuristic and machine learning for slope stability assessment: A multinational data analysis", *Expert Syst. Appl.*, **46**, 60-68.

Ivakhnenko, A. (1971), "Polynomial theory of complex systems", *IEEE Trans. Syst. Man Cybernet.*, (4), 364-378.

Ivakhnenko, A.G. and Ivakhnenko, G.A. (2000), "Problems of further development of the group method of data handling algorithms. Part I", *Pattern Recogn. Image Anal. C/C Raspoznavaniye Obrazov Analizizobrazhenii*, **10**(2), 187-194.

Kaveh, A., Bakhshpoori, T. and Hamze-Ziabari, S. (2017), *M5′ and Mars Based Prediction Models for Properties of Self-Compacting Concrete Containing Fly Ash*, in *Periodica Polytechnica Civil Engineering*.

Kaveh, A., Bakhshpoori, T. and Hamze-Ziabari, S.M. (2016), "Derivation of new equations for prediction of principal ground-motion parameters using M5′ algorithm", *J. Earthq. Eng.*, **20**(6), 910-930.

Kaveh, A., Bakhshpoori, T. and Hamze-Ziabari, S.M. (2017), "New model derivation for the bond behavior of NSM FRP systems in concrete", *Iran. J. Sci. Technol. Trans. Civ. Eng.*, **41**(3), 249-262.

Kaveh, A., Hamze-Ziabari, S.M. and Bakhshpoori, T. (2016), "Patient rule-induction method for liquefaction potential assessment based on CPT data", *Bull. Eng. Geol. Environ.*, 1-17.

Kaveh, A., Hamze-Ziabari, S.M. and Bakhshpoori, T. (2017), "M5' algorithm for shear strength prediction of HSC slender beams without web reinforcement", *J. Model. Optim.*, **7**(1), 48-53.

Kostić, S., Vasović, N. and Sunarić, D. (2016), "Slope stability analysis based on experimental design", *J. Geomech.*, **16**(5), 04016009.

Lee, T.L., Lin, H.M. and Lu, Y.P. (2009), "Assessment of highway slope failure using neural networks", *J. Zhejiang Univ. Sci. A*, **10**(1), 101-108.

Li, J. and F. Wang (2010), *Study on the Forecasting Models of Slope Stability Under Data Mining*, in *Earth and Space: Engineering, Science, Construction, and Operations in Challenging Environments*, 765-776.

Lin, H.M., Chang, S.K., Wu, J.H. and Juang, C.H. (2009), "Neural network-based model for assessing failure potential of highway slopes in the Alishan, Taiwan Area: Pre- and post-earthquake investigation", *Eng. Geol.*, **104**(3-4), 280-289.

Lu, P. and Rosenbaum, M.S. (2003), "Artificial neural networks and grey systems for the prediction of slope stability", *Nat. Hazards*, **30**(3), 383-398.

Nannings, B., Abu-Hanna, A. and De Jonge, E. (2008), "Applying PRIM (Patient Rule Induction Method) and logistic regression for selecting high-risk subgroups in very elderly ICU patients", *J. Med. Inform.*, **77**(4), 272-279.

Quinlan, J.R. (1992), "Learning with continuous classes", *Proceedings of the 5th Australian Joint conference on Artificial Intelligence*, Hobart, Tasmania, Australia, November.

Sah, N.K., Sheorey, P.R. and Upadhyaya, L.N. (1994), "Maximum likelihood estimation of slope stability", *J. Rock Mech. Min. Sci. Geomech. Abstr.*, **31**(1), 47-53.

Samui, P. (2008), "Slope stability analysis: A support vector machine approach", *Environ. Geol.*, **56**(2), 255-267.

Samui, P. (2011), "Utilization of relevance vector machine for rock slope stability analysis", *J. Geotech. Eng.*, **5**(3), 351-355.

Samui, P. and Kothari, D.P. (2011), "Utilization of a least square support vector machine (LSSVM) for slope stability analysis", *Scientia Iranica*, **18**(1), 53-58.

Samui, P., Lansivaara, T. and Kim, D. (2011), "Utilization relevance vector machine for slope reliability analysis", *Appl. Soft Comput.*, **11**(5), 4036-4040.

See, L. and Openshaw, S. (1999), "Applying soft computing approaches to river level forecasting", *Hydrol. Sci. J.*, **44**(5), 763-778.

Sinha, A.K. and Sengupta, M. (1989), "Expert system approach to slope stability", *Min. Sci. Technol.*, **8**(1), 21-29.

Wang, H.B., Xu, W.Y. and Xu, R.C. (2005), "Slope stability evaluation using back propagation neural networks", *Eng. Geol.*, **80**(3-4), 302-315.

Wang, Y. and Witten, I.H. (1996), *Induction of Model Trees for Predicting Continuous Classes*, University of Waikato, Hamilton, New Zealand.

Witten, I.H. and Frank, E. (2005), *Data Mining: Practical machine Learning Tools and Techniques*, Morgan Kaufmann.

Xiaoming, Y. and Xibing, L. (2011), "Bayes discriminant analysis method for predicting the stability of open pit slope" *Proceedings of the 2011 International Conference on Electric Technology and Civil Engineering (ICETCE)*, Lushan, China, April.

Zhang, Z., Liu, Z., Zheng, L. and Zhang, Y. (2014), "Development of an adaptive relevance vector machine approach for slope stability inference", *Neural Comput. Appl.*, **25**(7), 2025-2035.

Zhao, H., Yin, S. and Ru, Z. (2012), "Relevance vector machine applied to slope stability analysis", *J. Numer. Anal. Meth. Geomech.*, **36**(5), 643-652.

Zhou, K.P. and Chen, Z.Q. (2009), "Stability prediction of tailing dam slope based on neural network pattern recognition", *Proceedings of the 2nd International Conference on Environmental and Computer Science*, Dubai, U.A.E., December.

*CC*