RESEARCH ARTICLE

Slope Stability and Safety Assessment Based on Random Forest Enhanced under Multi-Strategy Pelican Optimization

Huanhuan Zhao[1], Xiaole Zhao[1], Xueyi Wen[1], Rongfang Yan[1,2] and Jiandong Zhang[1,2,*]

[1] College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, China
[2] Gansu Provincial Research Center for Basic Disciplines of Mathematics and Statistics, Lanzhou 730070, China

Abstract

To improve the prediction accuracy of slope stability and prevent slope failure accidents, this study proposes a slope stability prediction model based on an improved pelican optimization algorithm optimized random forest (Improved Pelican Optimization Algorithm optimized Random Forest, IPOA-RF). First, according to 431 slope cases, the slope height, slope angle, unit weight, cohesion, internal friction angle, and pore water pressure ratio were selected as the main predictive features. Second, due to the issue of excessive hyperparameters in the traditional random forest (RF) model, the IPOA algorithm was employed to optimize the RF parameters using an optimal-guidance strategy, mutation operator, and dynamically adjusted search mechanism. Finally, compared with five other optimization algorithms, the proposed IPOA algorithm exhibited superior parameter optimization ability and convergence performance in ten benchmark test functions. The designed IPOA-RF model achieved an average prediction accuracy of 85.1%, approximately 10.4% higher than that of the traditional RF model (74.7%). The results demonstrate that the IPOA-RF model can rapidly and accurately identify slope stability conditions, effectively overcoming the limitations of conventional methods. This model not only provides an innovative solution for slope stability assessment but also offers technical support for enhancing the safety and operational efficiency of practical slope engineering projects.

Keywords: slope stability prediction, visualization analysis, RF, IPOA-RF.

1 Introduction

The mechanical equilibrium of slopes is often disturbed by natural or human activities, leading to slope instability and geological disasters such as landslides, which can cause severe casualties and economic losses. In open-pit mining areas, slope instability is among the most critical and hazardous issues. Slope stability not only serves as a key indicator ensuring the safety of mountain engineering but also plays an important role in disaster prevention and mitigation by reducing landslide risks and protecting personnel and property [1]. Therefore, accurately predicting slope stability and

Submitted: 15 February 2026
Accepted: 12 March 2026
Published: 03 April 2026

Vol. 2, **No.** 2, 2026.
10.62762/TSSR.2026.963232

***Corresponding author:**
Jiandong Zhang
jdzhang@nwnu.edu.cn

Citation
Zhao, H., Zhao, X., Wen, X., Yan, R., & Zhang, J. (2026). Slope Stability and Safety Assessment Based on Random Forest Enhanced under Multi-Strategy Pelican Optimization. *ICCK Transactions on Systems Safety and Reliability*, 2(2), 82–100.

© 2026 ICCK (Institute of Central Computation and Knowledge)

preventing slope failure have become vital research topics in the field of geotechnical engineering. At present, slope stability analysis methods mainly include limit equilibrium theory and numerical simulation models. Azarafza et al. [2] summarized the applications of limit equilibrium theory in slope stability research, but these methods rely heavily on sliding surface assumptions and contain a certain degree of subjectivity [3]. Ureel et al. [4] pointed out that when dealing with complex geological environments, traditional numerical simulation methods are highly dependent on model structure and boundary conditions. Consequently, such methods require extensive engineering experience to yield reasonable analytical results. Li et al. [5] further showed that numerical models involve complex computational processes and high computational costs, limiting their applicability for rapid stability prediction. Hence, while traditional approaches remain widely used in engineering practice, their data requirements, computational complexity, and real-time performance still pose limitations.

With the development of data-driven methods, machine learning has become an important tool for slope stability research. Gordan et al. [6] constructed a combined model integrating artificial neural networks (ANN) with particle swarm optimization (PSO) for sensitivity analysis, achieving promising results. Chakraborty et al. [7] and Kasa et al. [8] analyzed 200 slope cases and developed models based on multiple linear regression (MLR) and ANN, demonstrating that both methods performed well in slope stability prediction. Huang et al. [9] introduced PSO into a k-nearest neighbor (KNN) model to improve predictive accuracy. Zhang et al. [10] proposed an XGBoost-based model for slope stability prediction and verified its high performance through multiple comparative experiments. Demir et al. [11] applied random forest (RF) as a classification tool and confirmed its strong generalization ability and robustness compared with traditional methods. Kurnaz et al. [12] compared three machine learning models—logistic regression, ANN, and random forest—showing that ANN and RF performed better based on RMSE and VAF indicators. Huang et al. [13] used 396 slope samples to establish a random forest-based model and confirmed its accuracy and efficiency in complex geological environments. Similarly, Rukhaiyar et al. [14] developed a hybrid model combining ANN and heuristic algorithms, achieving an overall prediction accuracy exceeding 91.1%.

Compared with existing hybrid optimization-based models for slope stability prediction, the main contributions of this study can be summarized as follows:

1. An improved Pelican Optimization Algorithm (IPOA) is proposed by introducing logistic chaotic mapping, opposition-based learning, and dynamic core-domain population adjustment, which effectively enhances both global exploration and local exploitation capabilities.
2. A systematic hyperparameter optimization framework is constructed by integrating IPOA with the Random Forest classifier, forming a novel IPOA-RF model for slope stability prediction.
3. The proposed method is comprehensively validated using benchmark functions, multiple metaheuristic algorithms, and comparative machine learning experiments, demonstrating superior stability and prediction performance. These contributions distinguish the proposed IPOA-RF framework from existing RF-based hybrid optimization models.

In summary, although both domestic and international scholars have achieved considerable progress in slope stability prediction, challenges remain, including insufficient model generalization ability, limited hyperparameter optimization efficiency, and unstable prediction outcomes. Therefore, this study selects 431 slope cases as research samples, extracts the main characteristic parameters, and verifies the data structure's rationality through correlation and visualization analysis. An Improved Pelican Optimization Algorithm (IPOA) is employed to optimize random forest (RF) model hyperparameters, constructing a slope stability prediction model (IPOA-RF). Through cross-validation and model comparison experiments, the effectiveness of the proposed model in improving prediction accuracy and stability is verified, and the significance of characteristic variables is analyzed to provide theoretical support for slope disaster prevention and early warning.

2 Methods

2.1 Data and Features

2.1.1 *Selection of Predictive Indicators*

By thoroughly analyzing various factors that influence slope stability and combining them with the

mechanisms of slope failure, as well as referring to previous research, this study identifies the key factors that should be considered in slope stability analysis. These include the physical strength of slope soil and rock, the geometric characteristics of the slope, and the influence of groundwater. Ultimately, six categories of indicators were selected as the input variables for slope stability prediction: slope height H, slope angle α, unit weight γ, cohesion c, internal friction angle φ, and pore water pressure ratio ru. A brief description of each input variable is provided in Table 1.

Table 1. Brief descriptions of predictive indicators.

Indicator	Description
Slope height (H)	The vertical distance from the slope toe to the slope crest.
Slope angle (α)	The angle between the inclined slope surface and the horizontal plane.
Unit weight (γ)	The weight per unit volume of the soil/rock mass.
Cohesion (c)	The component of shear strength that is independent of the effective normal stress during soil deformation.
Internal friction angle (φ)	A soil parameter representing the material's ability to resist shear due to interparticle friction.
Pore water pressure ratio (ru)	The ratio of pore water pressure to the overburden (vertical) stress, i.e., $ru = u/\sigma_v$.

2.1.2 Data

The dataset used in this study was obtained from previous literature [15, 16], comprising a total of 431 slope cases. In the dataset, unstable slopes are labeled as "1," whereas stable slopes are labeled as "0." It is noteworthy that the dataset contains 285 stable slopes and 146 unstable slopes.

2.2 Machine Learning Models

In the field of machine learning, different algorithms are suited to different tasks and data types. To comprehensively evaluate the predictive performance of various algorithms for slope stability, this study selects six representative models: Random Forest (RF), Support Vector Machine (SVM), Long Short-Term Memory network (LSTM), Convolutional Neural Network (CNN), Extreme Gradient Boosting (XGBoost), and k-Nearest Neighbors (KNN). These models span ensemble learning, kernel methods, deep learning, and instance-based learning, enabling a comparative analysis of predictive performance and model capability from different perspectives.

2.2.1 Random Forest

Random Forest (RF) is a powerful and widely used machine learning method. It combines ensemble learning and decision trees, providing a powerful and general-purpose tool for handling complex classification tasks. Its core idea is to build multiple decision trees and aggregate their prediction results (such as voting or averaging), thereby obtaining the final prediction result. The randomness introduced during the tree construction ensures diversity among trees, reduces overfitting, and improves the generalization ability of the model [17].

Random Forest has also been widely applied in the field of geotechnical engineering. Previous studies have shown that Random Forest can effectively handle nonlinear, high-dimensional, and noisy data commonly encountered in geotechnical engineering, and has therefore been used for various prediction and analysis tasks. It has also been applied to the inversion and identification of soil parameters, using existing monitoring or experimental data to back-calculate key mechanical parameters such as cohesion, internal friction angle, and permeability coefficient. Compared with traditional methods, Random Forest does not rely on complex numerical simulations or inversion calculations during the parameter inversion process, but instead uses a data-driven approach to automatically learn the relationships between inputs and outputs, thereby improving the reliability and stability of parameter prediction while reducing computational cost. Therefore, Random Forest has become an important intelligent prediction tool in geotechnical engineering.

2.2.2 Support Vector Machine Model

Support Vector Machine (SVM) was proposed by Corinna Cortes and Vapnik in 1995 [24], it improves the computational power of the algorithm by adopting the principle of structural risk minimization, and finds the optimal separating hyperplane based on this principle, using the hyperplane to achieve better classification performance. The basic idea is to map low-dimensional nonlinear sample data into a high-dimensional space through kernel functions, and to search for a hyperplane with the maximum margin in this space, thereby constructing the optimal decision function, and classifying different categories of samples; it exhibits unique advantages in handling problems such as small sample sizes, nonlinearity, and avoiding local optima [18].

2.2.3 LSTM Model

Long Short-Term Memory (LSTM) network is a special type of Recurrent Neural Network (RNN), traditional RNN has short-term memory limitations, when processing long sequences, early information gradually fades, and all input information is treated equally, making it unable to selectively remember important information, leading to insufficient filtering of critical information [19]. LSTM improves upon RNN by adding a gating mechanism to control information forgetting and flow, and simultaneously introduces a cell state, which acts like an information conveyor belt, allowing information to be transmitted and stored throughout the sequence, enabling LSTM to capture and process long-term dependencies, and avoiding the vanishing or exploding gradient problems of basic RNN. LSTM is typically formed by stacking multiple network layers, each layer has its own hidden and cell states, which enables the network to have stronger representational power, and to learn more complex feature representations.

2.2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning algorithm, widely used in fields such as image and speech processing, it is a variant of artificial neural networks, particularly suitable for processing data with grid structures, its core idea is to use convolutional layers to automatically extract data features, and gradually learn more complex patterns through multiple network layers. By relying on local connections and weight sharing mechanisms, CNN can efficiently utilize local information, on the one hand, it reduces the number of model parameters and lowers computational complexity, on the other hand, it enhances the generalization ability of the model, and parameter sharing helps reduce the risk of overfitting, while also saving a large amount of memory and computational resources.

2.2.5 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an advanced supervised learning algorithm proposed by Chen and Guestrin [25], based on decision tree machine learning methods. To overcome the limitations of a single decision tree, XGBoost uses a multiple decision-tree-based machine learning model, establishes a mapping from input to output, and is used to explore the potential relationships between geotechnical parameters and the responses of interest. Its uniqueness lies in leveraging the advantages of multiple trees, which improves modeling efficiency, and this model has been proven effective in slope stability analysis.

2.2.6 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a classical instance-based supervised learning method in machine learning, its core idea is that, for a target sample, the algorithm measures its distance to all training samples (such as Euclidean distance), selects the K nearest categories, and performs label or value estimation to complete the prediction task. KNN provides a data-driven and nonparametric method for slope stability analysis, and is particularly suitable for rapid evaluation under complex geological conditions.

2.3 Pelican Optimization Algorithm (POA)

2.3.1 Overview of the POA Algorithm

The Pelican Optimization Algorithm (POA) was proposed by Trojovsky et al. [20]. It is a novel nature-inspired metaheuristic algorithm that simulates the cooperative hunting behavior of pelicans in search of prey. The core idea of POA is to model the pelican's diving and attacking process during hunting to update individual positions and iteratively approach the global optimum. Owing to its simple structure and strong exploration ability, POA has attracted increasing attention in recent years.

2.3.2 Mathematical Modeling of the Algorithm

The initialization process of pelican positions can be expressed as:

$$X_{i,j} = l_j + rand(u_j - l_j), \quad i = 1, 2, \ldots, N;\ j = 1, 2, \ldots, m \tag{1}$$

where $X_{i,j}$ represents the position of the i-th pelican in the j-th dimension, N is the number of pelicans, m denotes the problem dimension, and $rand$ is a uniformly distributed random variable in $[0, 1]$. Parameters u_j and l_j represent the upper and lower bounds of the search space for dimension j.

In the Pelican Optimization Algorithm (POA), the pelican population can be represented by the following population matrix:

$$X = \begin{bmatrix} X_1 \\ X_i \\ X_j \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,m} \end{bmatrix}_{N\times m}. \tag{2}$$

where X represents the population matrix of pelicans, and X_i denotes the position of the i-th pelican.

$$F = \begin{bmatrix} F_1 \\ F_i \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} . \quad (3)$$

where F represents the objective function vector of the pelican population, and F_i denotes the objective function value of the i-th pelican.

In the first stage, pelicans determine the location of their prey and then move toward this fixed region. This stage constructs a neighborhood around the prey to enable the POA algorithm to explore and exploit different regions of the search space. Similar to the behavior of the Pelican Defense Algorithm (PDA), the POA algorithm can generate new exploration points in the search space. A key characteristic of the POA algorithm is that the prey's location in the search space is generated randomly, which enhances the algorithm's ability to escape local optima and improves its exploration capability. For each pelican, the updating strategy is defined as follows:

$$X_{i,j}^{P_1} = \begin{cases} X_{i,j} + rand \cdot (P_j - I \cdot X_{i,j}), & F_p < F_i \\ X_{i,j} + rand \cdot (X_{i,j} - P_j), & else \end{cases} \quad (4)$$

where $X_{i,j}^{P_1}$ represents the updated position of the i-th pelican in the j-th dimension during the first stage, $rand$ is a random number uniformly distributed in $[0, 1]$, I is a random integer equal to 1 or 2, P_j denotes the prey's position in the j-th dimension, and F_p is the objective function value of the prey.

In the POA algorithm, if the objective function value at the new position improves, the pelican accepts the new position. This type of update is called a valid update, which ensures that the algorithm does not move toward suboptimal regions. The update process can be expressed as:

$$X_i = \begin{cases} X_i^{P_1}, & F_i^{P_1} < F_i \\ X_i, & else \end{cases} \quad (5)$$

where $X_i^{P_1}$ represents the new position of the i-th pelican after the first-stage update, and $F_i^{P_1}$ denotes the corresponding objective function value.

In the second stage, when pelicans reach the water surface, they spread their wings on the surface and then place their prey into their throat pouch. This flying behavior allows pelicans to obtain more prey from the attacked region. To simulate this process, the POA algorithm uses this behavior to update pelicans' positions to better ones, thereby enhancing its local search and exploitation capabilities. Mathematically, this process can be described as follows:

$$X_{i,j}^{P_2} = X_{i,j} + R \cdot \left(1 - \frac{t}{T}\right) \cdot (2 \cdot rand - 1) \cdot X_{i,j} \quad (6)$$

where $X_{i,j}^{P_2}$ represents the updated position of the i-th pelican in the j-th dimension during the second stage, $rand$ is a random number in the range $[0, 1]$, R is a random integer equal to 1 or 2, t is the current iteration number, and T is the maximum number of iterations.

If the objective function value at the new position is improved, the pelican updates its position according to the following mathematical model:

$$X_i = \begin{cases} X_i^{P_2}, & F_i^{P_2} < F_i \\ X_i, & else \end{cases} \quad (7)$$

where $X_i^{P_2}$ represents the new position of the i-th pelican, and $F_i^{P_2}$ denotes the objective function value at this new position $X_i^{P_2}$ after the second stage update.

2.4 Improvements of the POA Algorithm

Although the POA algorithm exhibits excellent stability and high convergence accuracy, it still suffers from the following shortcomings:

1. During the population initialization process, the distribution of initial solutions is random and non-uniform. In addition, the quality of individuals in the population is also uneven, which can easily lead to the loss of population diversity and may cause the algorithm to miss the potential optimal solution.

2. In the second stage of POA, pelicans rotate around their current positions and randomly select new positions in the search space. As the number of iterations increases, the ability to escape local optima gradually weakens, and the current positions of the population determine the search domain. This can easily lead the algorithm into local optima, making it difficult for the population to find the global optimum.

2.4.1 Initialization Stage Improvement

In the population initialization process of POA, the uneven distribution of individuals may cause premature convergence. To address this problem, this paper introduces a hybrid mapping and opposition-based learning mechanism based on random initialization. First, the initial individuals are generated uniformly as follows:

$$Z_i \sim U(l_j, u_j), \quad i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, m$$

where u_j and l_j represent the upper and lower bounds of the j-th dimension of the problem, respectively, and m is the problem dimension.

For each dimension, Logistic mapping is used to enhance adaptability:

$$z_{i,j}^{(t+1)} = u z_{i,j}^{(t)} \left(1 - z_{i,j}^{(t)}\right), \tag{8}$$

where $z_{i,j}^{(t)}$ represents the state of the i-th individual in the j-th dimension at iteration t, and $z_{i,j}^{(t+1)}$ represents the updated state after Logistic mapping.

The parameter μ is the control parameter of Logistic mapping. When $\mu = 4$, the sequence reaches a completely chaotic state, which enables the generation of sequences with good ergodicity and uniform distribution in the interval $[0, 1]$. Then, opposition-based learning is performed as follows:

$$\tilde{z}_{i,j} = l_j + u_j - z_{i,j}^{(1)} \tag{9}$$

where l_j and u_j represent the lower and upper bounds of the j-th dimension of the search space, respectively, and $\tilde{z}_{i,j}$ denotes the opposite solution of the i-th individual in the j-th dimension.

Finally, individuals with better fitness are selected from $\{z_{i,j}^{(1)}, \tilde{z}_{i,j}\}$ as the initial solutions to be incorporated into the population, thereby improving the uniformity and diversity of the initial population distribution.

2.4.2 Dynamic Population Adjustment Based on Core Domain

During the iteration process, POA tends to suffer from premature convergence to local optima, which leads to a decrease in global exploration ability. To address this issue, this study introduces a dynamic adjustment mechanism based on the core domain. First, the initial radius between the population center and the global optimum is defined as follows:

$$R_0 = \left\| X^{best} - \bar{X} \right\|_2, \qquad \bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$$

where R_0 represents the Euclidean distance between the current global optimal solution and the population center, serving as the initial radius. X^{best} denotes the current global optimal solution, $\bar{X}$ represents the population center, and N is the population size.

The radius is dynamically adjusted during iterations using an adaptive contraction coefficient:

$$R_t = \kappa \, c(t) \, R_0 \tag{10}$$

$$c(t) = \frac{1}{1 + \exp\left[-f\left(X^{best}\right)\right]}$$

where $c(t)$ is the adaptive coefficient with a value in the range $(0, 1)$, used to dynamically adjust the radius, and is defined as: $f(X^{best})$ represents the fitness function value of the global optimal solution at iteration t, and κ is the radius scaling factor, which is typically set to $\kappa = 1.5$.

When the number of individuals within the core domain becomes too large, the algorithm performs a sparsification operation in this region and replenishes new individuals in the outer region. This process helps maintain the diversity and uniform distribution of the population, thereby preventing premature convergence and preserving the algorithm's global exploration capability. The triggering condition is defined as follows:

$$\left| \left\{ \left\| X_i - X^{best} \right\| \le R_t \right\} \right| > \theta N \tag{11}$$

where θ represents the ratio threshold of the core domain. If the number of individuals within the core domain exceeds θN, the sparsification process is triggered.

2.4.3 Improvement of the Exploitation Phase

The traditional POA algorithm tends to search for better solutions near the current solution during the attacking and escaping phases, but premature exploitation often causes the algorithm to fall into local optima, particularly in complex boundary data regions where this issue becomes more pronounced. As the

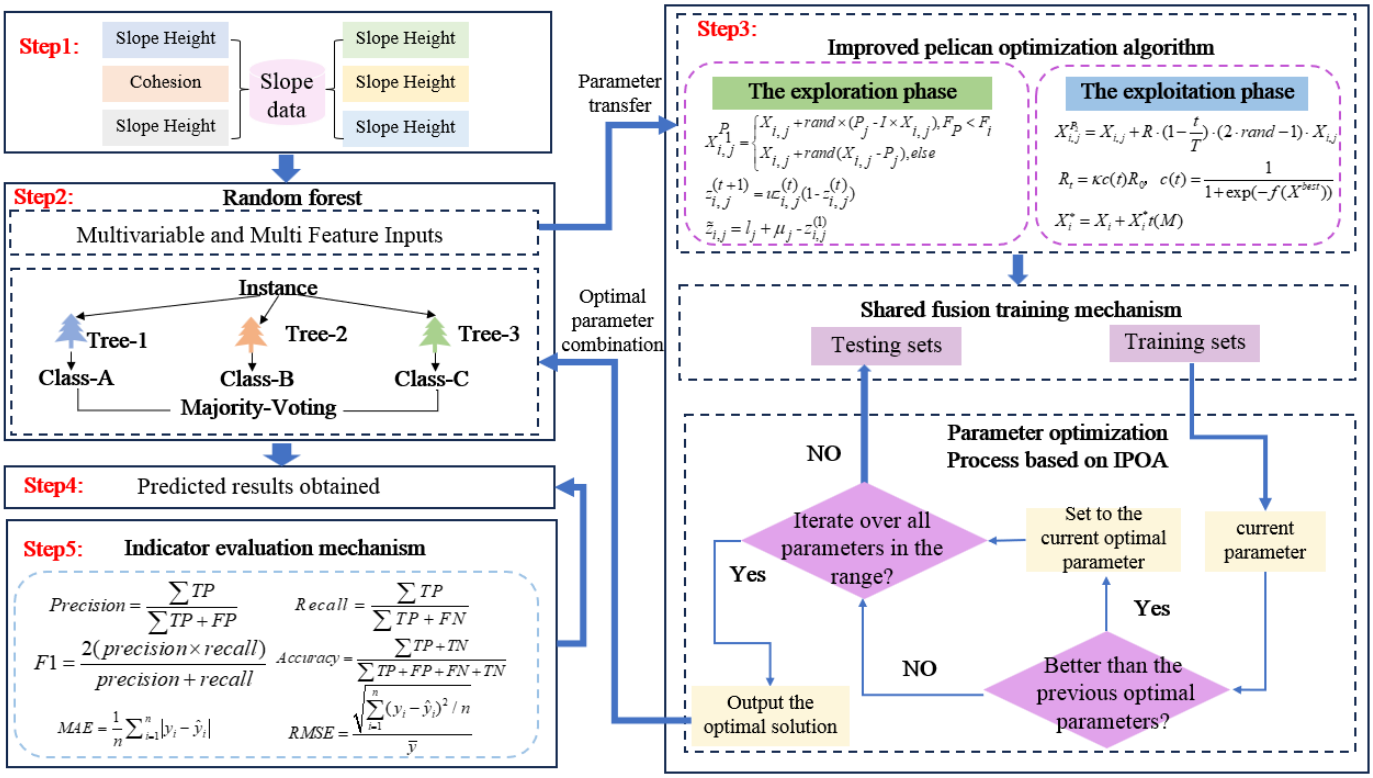


Figure 1. Overall framework of the IPOA–RF model.

number of iterations increases, the search particles tend to cluster around the current solution. At this stage, improving the solution quality requires a more refined local search within a smaller region. Therefore, to address this issue, this study introduces random perturbations through variation operators during the exploitation phase, enabling individuals to jump away from the current local optimum, thereby increasing the likelihood of exploring the global optimum again. As the number of iterations grows, the updating strategy can guide the search toward more promising regions, thereby improving the quality of the solutions. The update rule is given by:

$$X_i^* = X_i + X_i^*(t(M)) \tag{12}$$

where X_i^* represents the position of the mutant pelican, X_i is the position of the i-th pelican, and $t(M)$ denotes a t distribution. The degrees of freedom depend on the number of POA iterations. In the early iterations, when the degrees of freedom are small, the t distribution exhibits a heavy-tailed distribution, which allows individuals to perform large-scale jumps, thereby enhancing the global exploration ability. In the later iterations, the degrees of freedom gradually increase, and the distribution gradually approaches a Gaussian distribution, thereby improving the local exploitation ability and ensuring convergence accuracy.

2.5 Construction of the IPOA-RF Model

To address the problem of low prediction stability and accuracy in slope stability prediction models caused by the difficulty of hyperparameter tuning in the RF model, the IPOA algorithm, which has a simple structure and relatively high accuracy, is adopted to perform hyperparameter optimization for the RF model. The IPOA–RF is shown in Algorithm 1.

Due to the fact that the number of decision trees and the maximum number of features directly separated have a direct impact on the classification performance and stability of the RF model, the IPOA algorithm is adopted to perform hyperparameter tuning on the RF model, so as to improve the prediction accuracy of the model. In order to accurately predict the stability of open-pit mine slopes, this paper proposes a prediction method based on IPOA–RF. Figure 1 shows the specific structure of each functional module in the IPOA–RF method.

1. The slope dataset is analyzed and normalized [21], and the dataset is randomly divided into training and testing sets according to the ratio of 8:2;
2. The parameters of the RF algorithm are initialized, and the search range of the hyperparameters to be tuned is determined;
3. The parameters of the IPOA algorithm are

Algorithm 1: IPOA–RF Algorithm

Initialization:
Initialize population using logistic-chaotic opposition-based learning strategy
Evaluate fitness and select current global best solution

For $t = 1$ to T:
For $i = 1$ to N:
Stage 1: Exploration (move toward prey)
For $j = 1$ to m:
Compute new state using Eq. (4)
End
Update i-th individual using Eq. (5)
Stage 2: Exploitation (foraging on water surface)
For $j = 1$ to m:
Compute new state using Eq. (6)
End
Update i-th individual using Eq. (7)
End
Perform dynamic adjustment using Eq. (10)
If core region population $> \theta N$ **then**
Adjust population position using Eq. (12)
End
Update position if better than previous
End

Output global best solution
Train Random Forest model with optimal parameters

initialized, the population size is set, and the position update is performed according to Equations (1)–(12). Five-fold cross-validation is conducted on the training set [22] to serve as the fitness function to verify the fitness of each individual;

4. When the maximum number of iterations is reached, the optimal hyperparameters are output, and the optimal IPOA–RF slope prediction model is constructed. Finally, the testing set is used to test the constructed model.

5. The performance of the IPOA–RF prediction model is evaluated, and the underlying application principles are explained in detail.

3 Experimental Design

3.1 Validation Protocol

To ensure the robustness and fairness of the experimental evaluation, the dataset was randomly divided into training and testing sets with a ratio of 8:2 using stratified sampling to preserve the original class distribution.

All hyperparameter optimization processes were conducted exclusively on the training set. Specifically, five-fold cross-validation was applied within the training set to evaluate candidate hyperparameter combinations during the IPOA optimization process, thus avoiding data leakage from the testing set.

3.2 Data Preprocessing

In order to ensure the success of data-driven projects, improve data quality, enhance model performance, and discover potential relationships between features, data preprocessing is an indispensable step. Since there are no duplicate values, missing values, or outliers in the dataset. Therefore, normalization processing is performed on the data.

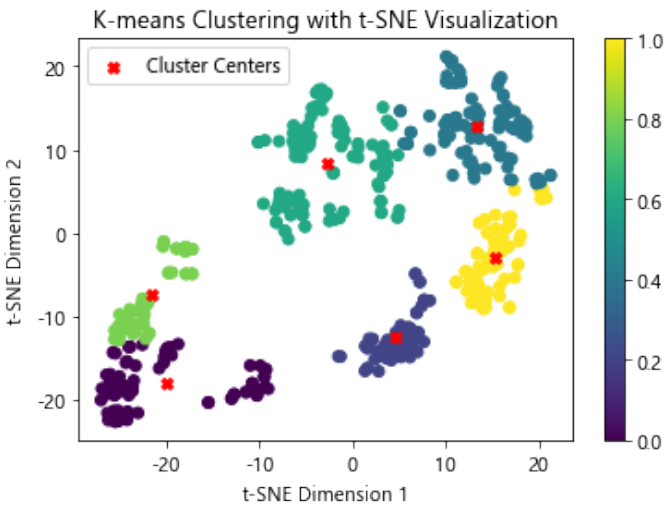


Figure 2. t-SNE Dimension Reduction — K-means clustering analysis.

The current data is processed using t-distributed stochastic neighbor embedding (t-SNE) to reduce the data to two dimensions, and the K-means algorithm is employed to facilitate easier identification of clusters and patterns. This method can also reduce unnecessary features, thereby decreasing noise, improving computational efficiency, and enhancing the interpretability of clustering results. In the two-dimensional space, the boundaries and feature variations of the clusters become clearer, which is helpful for quickly understanding the key

characteristics of the data. The results are visualized, as shown in Figure 2. It can be clearly observed that the data points are divided into six clusters of different colors, with points within the same cluster densely aggregated, and the distribution characteristics conform to the properties of the dataset. No small clusters that are far away from the main clusters appear, indicating that there are no abnormal values or special outliers, and the data preprocessing meets the requirements.

3.3 Hyperparameter Settings

To verify the effectiveness of the improved Pelican Optimization Algorithm (IPOA) proposed in this study, it is compared with four commonly used metaheuristic optimization algorithms, including Genetic Algorithm (GA), Differential Evolution (DE), Grey Wolf Optimizer (GWO), and the original Pelican Optimization Algorithm (POA). Each algorithm is independently executed 30 times, with a maximum of 500 iterations per run and a population size of $N = 40$. The specific parameter settings for each algorithm are listed in Table 2.

Table 2. Parameter settings of algorithms.

Algorithm	Main Parameters
IPOA	$\gamma = 0.2,\ P = 0.5$
POA	$\gamma = 0.2$
GA	$P_c = 0.7,\ P_m = 0.1$
DE	$F = 0.5,\ CR = 0.5$
GWO	$a\ :\ [0, 2]$ linearly decreases within the range

3.4 Test Function Settings

A total of 10 benchmark test functions were selected for comprehensive evaluation. Among them, $F_1 \sim F_5$ are unimodal functions, which are used to evaluate the exploitation accuracy and convergence speed of the algorithms; $F_6 \sim F_{10}$ are multimodal functions, which contain multiple local optima and a single global optimum, and are used to evaluate the algorithms' exploration and development abilities. Following [23], the average best value and the standard deviation of the test results are used to reflect the stability and optimization performance of the algorithms. The detailed information of these test functions is presented in Table 3. The formulas are as follows:

$$avg = \frac{1}{N}\sum_{i=1}^{N} BF_i$$

$$std = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(BF_i - avg)^2}$$

where N denotes the number of iterations, and BF_i represents the best solution obtained in the i-th run.

Table 3. Information on test functions.

Function	Test Function	Search Range	Dimension	Optimum
F1	Sphere	$[-100, 100]^m$	30	0
F2	Schwefel2.22	$[-10, 10]^m$	30	0
F3	Schwefel1.2	$[-100, 100]^m$	30	0
F4	Schwefel2.21	$[-100, 100]^m$	30	0
F5	Quartic	$[-1.28, 1.28]^m$	30	0
F6	Rastrigin	$[-5.12, 5.12]^m$	30	0
F7	Ackley	$[-32, 32]^m$	30	0
F8	Griewank	$[-600, 600]^m$	30	1
F9	Branin	$x_1 \in [-5, 10], x_2 \in [0, 15]$	2	0.398
F10	Hartmans	$[0, 1]^3$	3	-3.862

3.5 Evaluation Metrics

In this study, accuracy, recall, F1-score, precision, and the Receiver Operating Characteristic curve (ROC) are adopted to conduct a comprehensive analysis and evaluation of the models, so as to compare the performance of different models. This facilitates selecting the optimal model for analyzing and predicting slope stability.

$$Precision = \frac{\sum TP}{\sum TP + FP}$$

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN}$$

$$F1 = \frac{2(precision \times recall)}{precision + recall}$$

$$Accuracy = \frac{\sum TP + TN}{\sum TP + FP + FN + TN}$$

$$RMSE = \frac{\sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2/n}}{\bar{y}}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where TP represents the number of samples that are predicted to be stable and actually stable; TN represents the number of samples that are predicted to be unstable and actually unstable; FP represents the number of samples that are predicted to be stable but actually unstable; FN represents the number of samples that are predicted to be unstable but actually stable; y_i is the true value; $\hat{y}_i$ is the predicted value.

4 Results and Discussion

4.1 Descriptive Statistics

To comprehensively characterize the statistical features of the factors influencing slope stability, this study conducts a distributional and correlation analysis on six key variables (slope height H, slope angle α, unit weight γ, cohesion c, internal friction angle φ, and pore water pressure ratio ru).

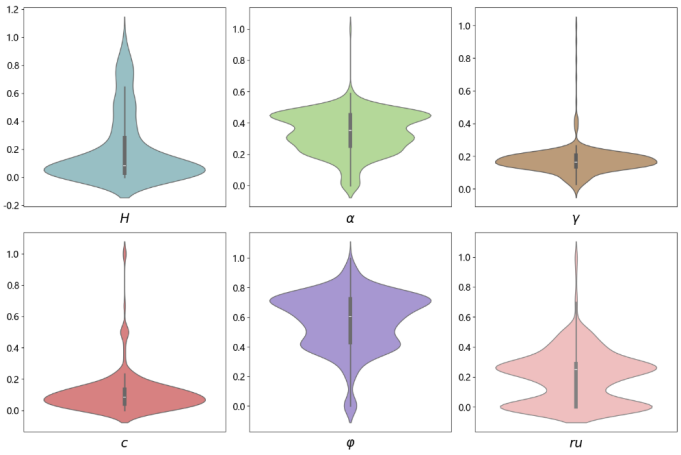


Figure 3. Violin plots of slope case distributions.

Figure 3 shows the kernel density distributions and violin plots of each variable. From the overall distribution, slope height, cohesion, and pore water pressure ratio all exhibit a clear right-skewed characteristic, indicating that most of the samples are concentrated in the relatively low-value region, while a certain number of extreme high values also exist. This reflects that these variables have a strong degree of dispersion and potential influence in the sample. In contrast, the distributions of slope angle and internal friction angle are more symmetric and concentrated in the middle region, indicating a relatively balanced distribution. The distribution of rock unit weight is relatively concentrated with a small degree of dispersion, suggesting that this variable is more stable in the sample.

Figure 4 further presents the scatterplot matrix of correlations between variables, showing the joint distribution characteristics of variables under stable and unstable slope conditions. Figure 5 uses the correlation heatmap to clearly identify the statistical correlations between variables. Under stable conditions, the correlation coefficient between slope height and slope angle is 0.40, indicating a moderate positive correlation; the correlation coefficient between cohesion and internal friction angle is 0.29, showing a certain degree of positive correlation between the two; and the correlation coefficient between slope height and pore water pressure ratio is –0.08, indicating a weak negative correlation. Under unstable conditions, the correlation between slope angle and internal friction angle increases significantly (correlation coefficient 0.56), indicating that the relationship between the two parameters is stronger under unstable conditions. In addition, the correlation between slope height and unit weight remains moderately positive (correlation coefficient 0.24), while the correlation between slope angle and pore water pressure ratio becomes slightly negative (correlation coefficient –0.19). These results reveal the differences in correlations between variables under stable and unstable slope states.

Figure 6 compares the distribution differences of various variables between stable and failed states from the perspectives of boxplots and histograms. The results indicate that under the stable state, slope height, cohesion, and pore water pressure ratio are concentrated at relatively low levels, with a limited number of outliers. In contrast, under the failed state, these variables show an overall increase in their average values, accompanied by a significant rise in the number of outliers, making the skewness characteristics of the distributions more pronounced.

Table 4. Mean and median values of variables.

Slope State	Feature	H	α	γ	c	φ	ru
Stable	Mean	140.43	34.70	24.29	37.49	37.49	0.30
Failed	Mean	67.49	31.54	21.19	34.92	36.00	0.26
Stable	Median	60.00	35.00	25.00	29.42	32.00	0.30
Failed	Median	50.00	30.00	20.20	20.00	30.00	0.25

Table 4 presents the statistical indicators of the variables. Overall, the differences between the stable and failed states are evident in both the mean and median values. Specifically, the mean values of slope height and pore water pressure ratio under the stable state are significantly higher, indicating their critical role in slope stability. Among these indicators, slope height exhibits the largest difference between the two

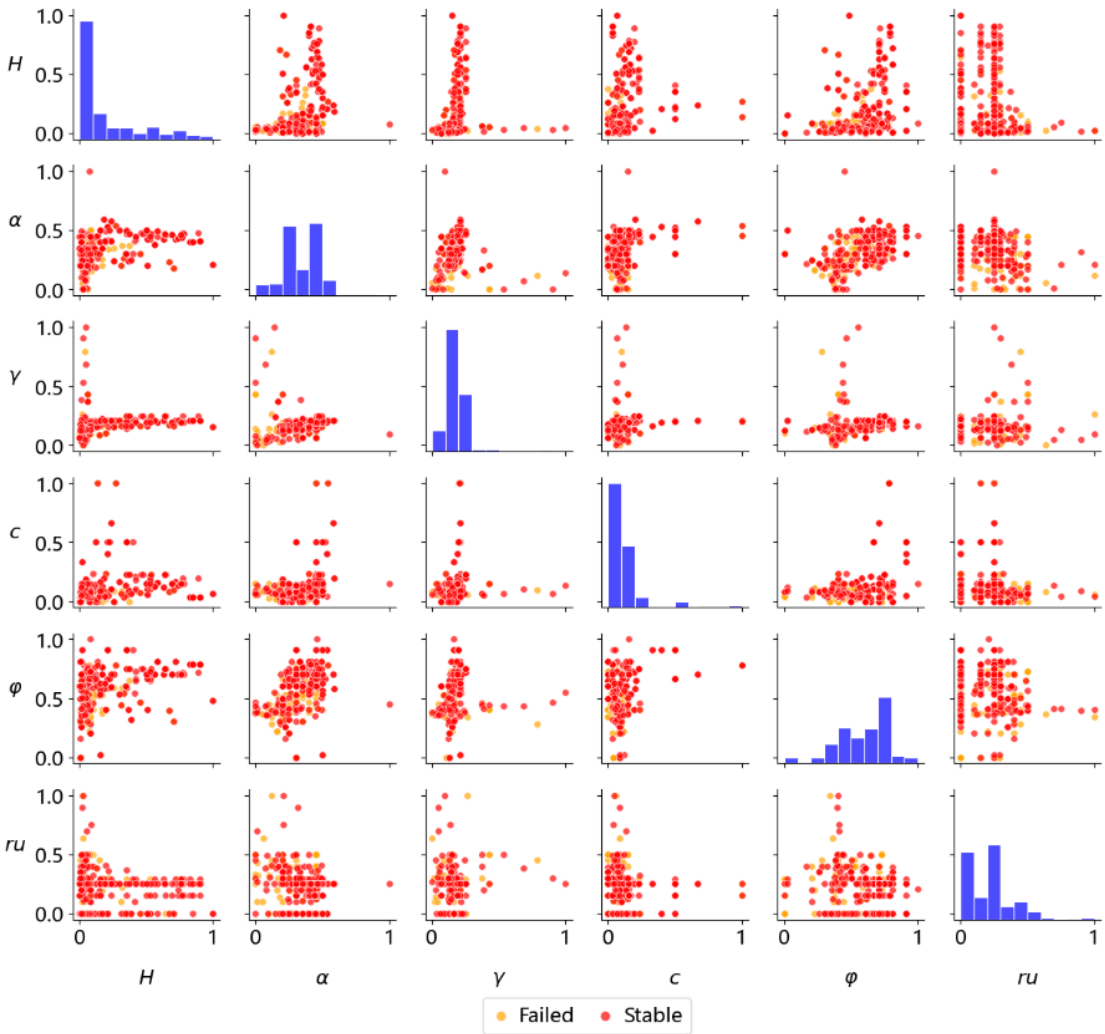


Figure 4. Correlation plot of variables.

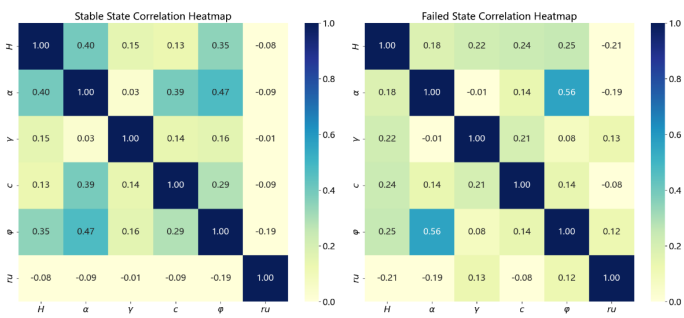


Figure 5. Correlation heatmaps of stable and failed slope states.

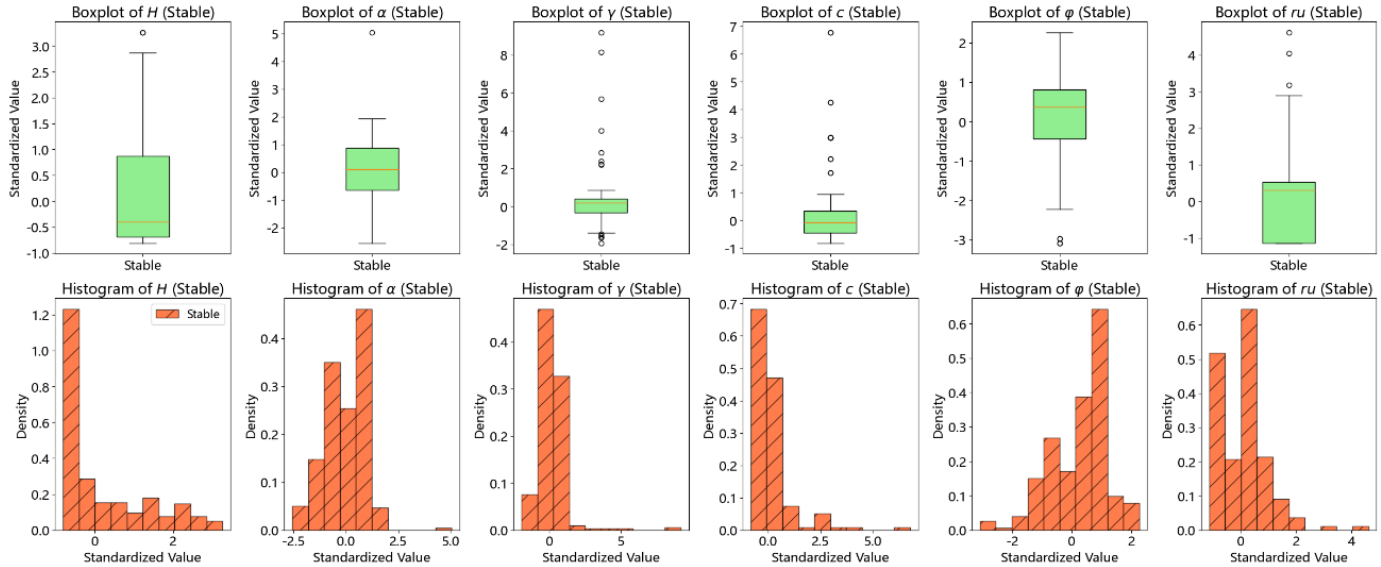
states. In the stable state, the mean and median values are 140.43 and 60.00, respectively, while in the failed state, they are 67.49 and 50.00, respectively. Overall, the mean and median values of the stability indicators are lower under the failed state than under the stable state.

Figure 7 further illustrates that the distributions of slope height and pore water pressure ratio under the failed state deviate more significantly from those under the stable state, indicating a stronger departure from normality. This suggests that extreme values of these variables play a critical role in slope failure.

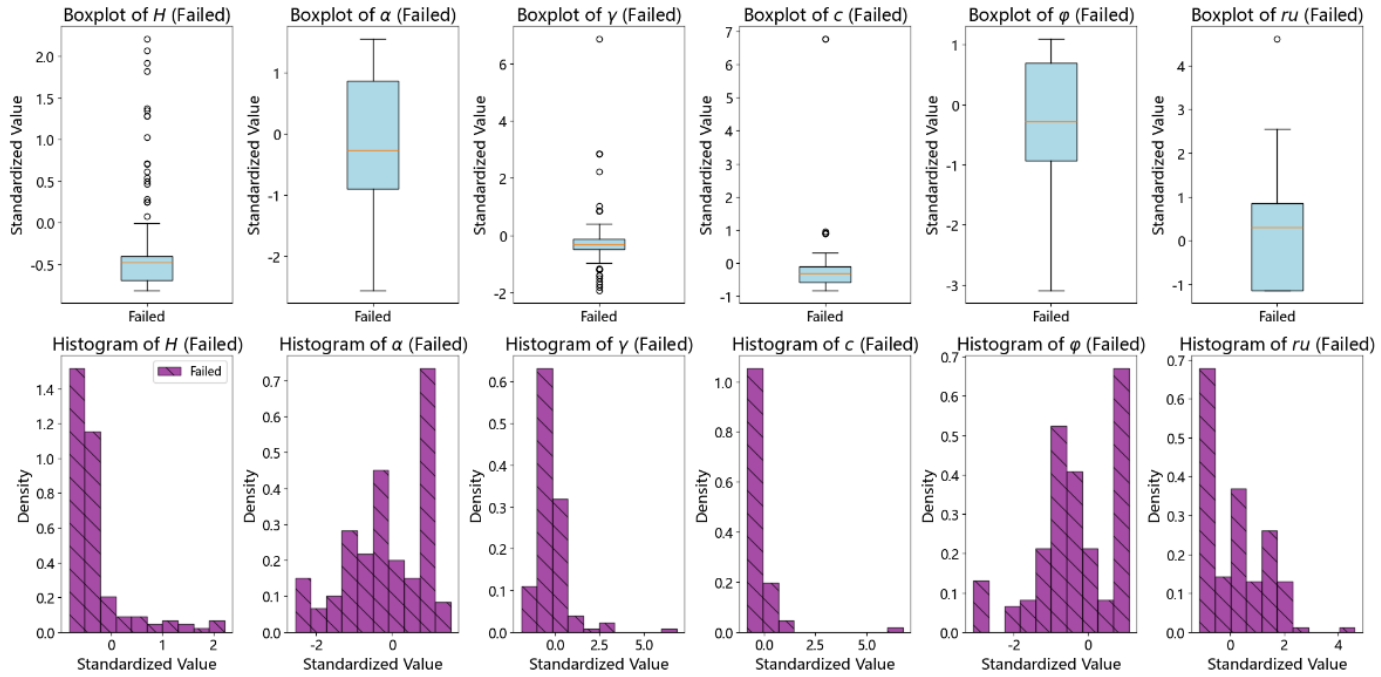
The Kolmogorov–Smirnov test (K–S test) is a goodness-of-fit test that uses sample data to infer whether the underlying population follows a specific theoretical distribution. It is suitable for examining the distribution of continuous random variables.

Table 5. Results of the K–S test.

Slope Stability Indicator	Statistic	P-value
H	0.2884	0.0000
α	0.1561	0.0157
γ	0.4142	0.0000
c	0.3429	0.0000
φ	0.2863	0.0000
ru	0.1520	0.0201



(a) Stable State



(b) Failed State

Figure 6. Boxplots and histograms of variables under stable and failed states.

The K–S test results in Table 5 indicate that most variables significantly deviate from the normal distribution, suggesting that these indicators exhibit non-normal characteristics in slope stability studies. Therefore, more robust statistical modeling methods are required to process such variables.

Based on the above analysis, slope height, pore water pressure ratio, slope angle, and internal friction angle are the main factors influencing slope stability. Among them, the first two variables show pronounced skewed distributions and greater dispersion under the failed state, while the latter two (α, φ) exhibit stronger correlations in the failed state. These findings provide a solid statistical foundation for subsequent feature selection and model construction using machine learning methods.

4.2 Comparison of Machine Learning Model Performance

According to the results of each evaluation index in Table 6, the Random Forest (RF) model performs excellently in terms of precision, F1-score, accuracy, and root mean square error (RMSE), especially

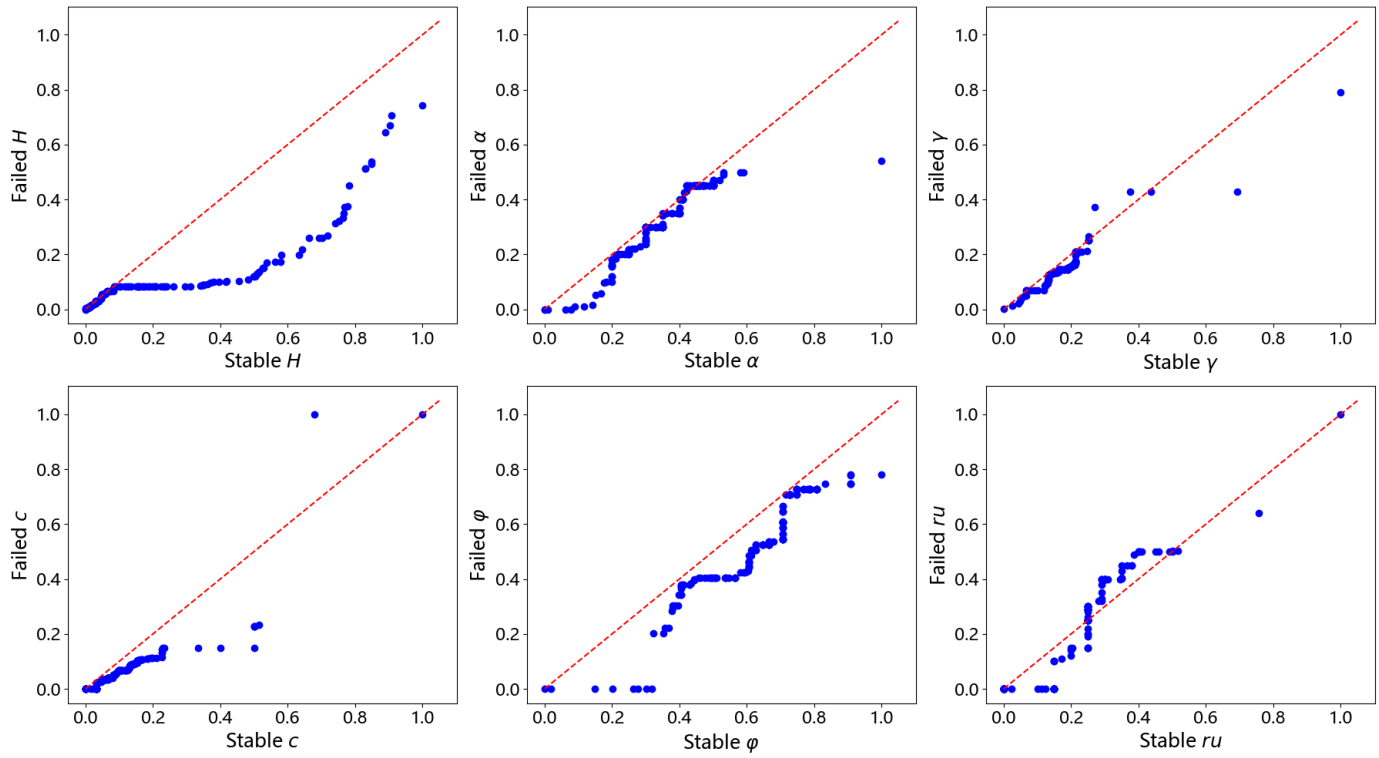


Figure 7. Q–Q plots of variables.

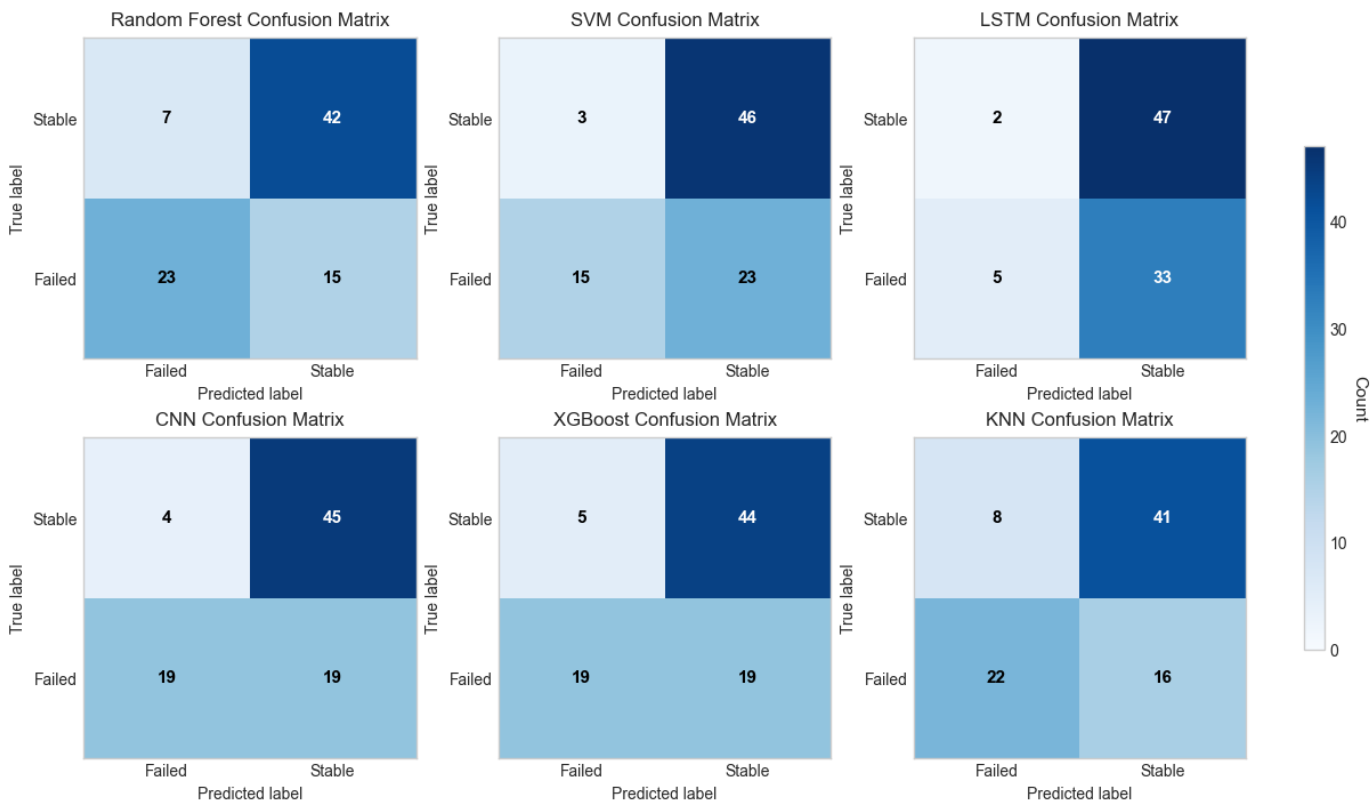


Figure 8. Comparison of Model Confusion Matrices.

showing outstanding advantages in precision and overall predictive accuracy. Although the Long Short-Term Memory (LSTM) network model has a relatively high recall rate, it performs poorly in precision, F1-score, accuracy, and RMSE indicators. Support Vector Machine (SVM), Convolutional Neural Network (CNN), and XGBoost improve the overall model performance to some extent. Therefore, we

Table 6. Performance evaluation of models.

Classifier	Assessment Score			Accuracy	RMSE	MAE
	Precision	Recall	F1-Score			
RF	0.74	0.86	0.79	0.7471	0.5029	0.2529
SVM	0.67	0.94	0.78	0.7011	0.5467	0.2989
LSTM	0.59	0.96	0.73	0.5977	0.6343	0.4138
CNN	0.70	0.92	0.79	0.7356	0.5142	0.2874
XGBoost	0.70	0.90	0.69	0.7241	0.5252	0.2759
KNN	0.72	0.84	0.77	0.7241	0.5252	0.2759

consider that the Random Forest model has the highest prediction accuracy for slope stability.

First, as shown in Figure 8, the performance of each model in predicting the "Stable" and "Failed" categories varies. When the true label is "Failed," the Random Forest model predicts the largest number of failed cases (23), while the LSTM model predicts the fewest failed cases (6), and the differences among the other four models are relatively small. When the true label is "Stable," the SVM and LSTM models predict the largest number of stable cases (46), whereas the KNN model predicts the fewest (41). Under this circumstance, the differences among the six models are not significant. Overall, the Random Forest model shows relatively balanced performance in predicting both categories, with fewer misclassifications. The performance of SVM, CNN, and XGBoost is relatively similar, with certain misjudgments. The LSTM model tends to misclassify "Failed" samples more frequently, indicating weaker overall performance. In summary, the Random Forest model achieves the best results in terms of predictive performance.

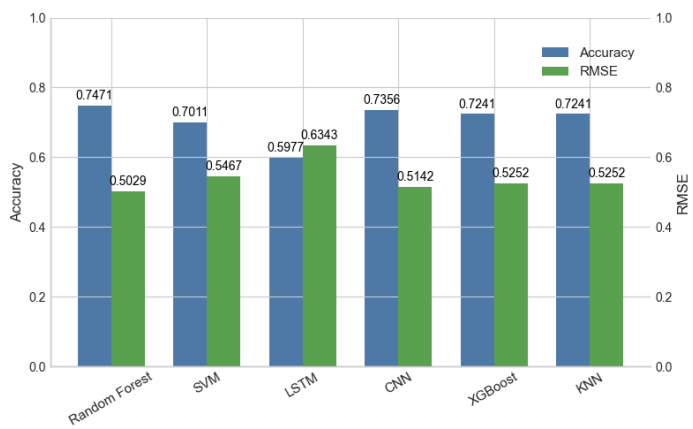


Figure 9. Comparison of Accuracy and RMSE among Six Models.

According to the comparison bar chart in Figure 9, it can be clearly seen that among the six models — Random Forest, SVM, LSTM, CNN, XGBoost, and KNN — the Random Forest model achieves the highest accuracy, reaching 0.7471. In contrast, the SVM model has a relatively low accuracy of 0.7011. The RMSE value of the LSTM model is 0.6343, which is relatively high among the six models, indicating that its prediction error is relatively large. The CNN, XGBoost, and KNN models show similar accuracy, averaging 0.7241, with comparable RMSE values of approximately 0.5252. Overall, the Random Forest model demonstrates the best comprehensive performance.

Based on the feature importance analysis results of the Random Forest model, it can be concluded that among the six key influencing factors, slope height and unit weight have the greatest impact on slope stability prediction results, followed by cohesion and density, while pore water pressure and internal friction angle have relatively smaller effects.

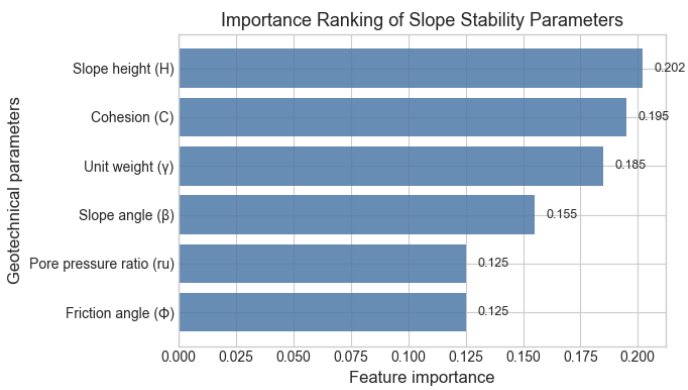


Figure 10. Importance ranking of slope stability parameters by Random Forest.

As illustrated in Figure 10, the feature importance analysis reveals that slope height (H) and unit weight

(γ) are the two most influential factors in slope stability prediction, with importance scores significantly higher than the other parameters. Cohesion (c) and internal friction angle (φ) exhibit moderate importance, while pore water pressure ratio (ru) and slope angle (α) contribute relatively less to the model's prediction decisions. This ranking provides valuable insights for prioritizing monitoring and reinforcement efforts in practical slope engineering applications.

This section focuses on the problem of slope stability prediction. By introducing six machine learning models, including Random Forest (RF), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM), a comparative analysis is conducted. A total of 431 groups of sample data are standardized and then divided into training and testing sets according to the ratio of 8:2. Based on precision, recall, root mean square error (RMSE), and mean absolute error (MAE), various indicators are used to evaluate the performance of each model. The results show that the Random Forest model performs the best overall, with an accuracy of 0.74, recall of 0.7471, and RMSE of 0.5029. In addition, in the confusion matrix analysis, the Random Forest model exhibits fewer misjudgments between the two slope states, demonstrating its feasibility and high precision in predicting slope stability, and providing a solid foundation for further research.

4.3 Performance Verification of IPOA

Table 7 presents the test results of five optimization algorithms. Functions F1–F5 are unimodal benchmark functions used to evaluate the local search ability of the algorithms, while functions F6–F10 are multimodal benchmark functions designed to assess global search capability. As shown in Table 7, the proposed IPOA algorithm achieves the best optimization performance on functions F1, F3, and F5. The results of the IPOA algorithm are more competitive and closer to the global optimum compared with other algorithms. Based on the results of the multimodal test functions, the IPOA algorithm also performs best on functions F6, F8, and F10, and exhibits global optimal convergence to zero on functions F5, F7, and F9. Since exploration and exploitation abilities are key indicators for successfully solving optimization problems, it can be concluded that the proposed IPOA method outperforms the previous algorithms overall.

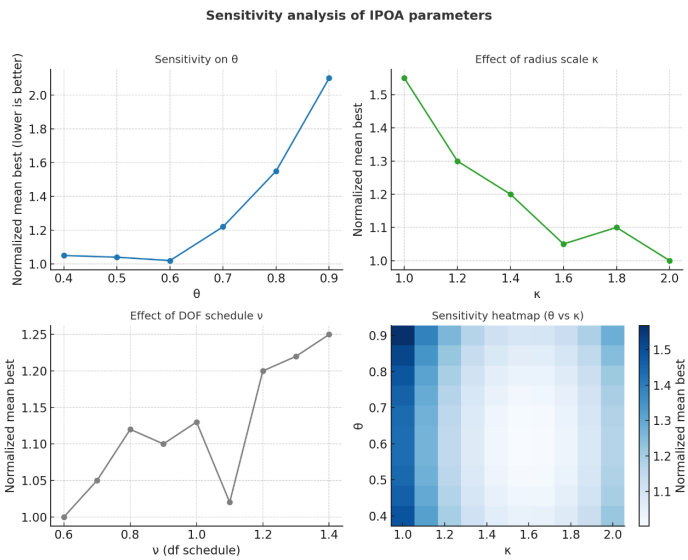


Figure 11. Sensitivity analysis of IPOA parameters.

As shown in Figure 11, the sensitivity analysis results of IPOA under different parameters θ, k, and the adaptive degree of freedom parameter v are illustrated. The results indicate that the algorithm performance

Table 7. Comparison of algorithm performance.

Algorithm	Statistic	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
IPOA	Mean	9.56E-152	2.12E-80	1.02E-151	6.85E-55	4.85E-285	5.64E-01	2.00E+01	5.29E-02	3.98E-01	-3.86E+00
	Std	4.80E-151	7.80E-80	3.44E-151	2.95E-54	0.00E+00	7.57E-01	2.00E+00	2.93E-02	3.98E-01	3.15E+00
	Best	1.03E-156	2.01E-82	1.53E-156	8.74E-63	3.16E-301	0.00E+00	2.00E+01	1.17E-07	3.98E-01	0.00E+00
POA	Mean	4.88E+01	3.90E+00	6.55E+01	4.74E+00	3.55E-03	6.35E+01	2.05E+01	1.53E+00	2.71E+01	-6.91E-02
	Std	4.13E+01	1.34E+00	2.86E+01	1.57E+00	5.26E-03	1.59E+01	1.23E+00	2.51E-01	2.77E+00	2.47E-02
	Best	1.40E+01	1.54E+00	9.50E-01	2.28E+00	1.03E-01	3.62E+01	3.63E+00	1.11E+00	1.31E+01	-1.31E+00
GA	Mean	2.38E+04	5.37E+05	5.26E+04	9.15E+01	4.40E+01	5.38E+03	2.21E+01	4.74E+02	3.05E+02	-1.23E-02
	Std	1.09E+03	1.04E+05	9.69E+02	6.32E+00	2.01E+00	9.21E+01	4.90E-02	1.15E+01	1.50E+01	1.48E-02
	Best	4.93E+04	2.42E+05	4.91E+04	7.61E+01	6.51E+01	5.23E+03	2.02E+01	4.46E+02	9.02E+01	-7.62E-02
DE	Mean	8.39E-37	4.91E-21	8.07E-37	3.70E-11	2.53E-73	3.00E+00	4.44E-16	1.28E-03	9.38E-01	-3.86E+00
	Std	5.44E-37	1.52E-21	5.39E-37	1.19E-11	2.29E-73	0.00E+00	0.00E+00	8.24E-04	0.00E+00	1.27E-15
	Best	1.26E-37	2.83E-21	7.66E-38	1.94E-11	1.99E-74	0.00E+00	4.44E-16	2.86E-04	2.93E-01	-3.86E+00
GWO	Mean	1.38E-203	3.87E-108	1.24E-201	1.45E-79	0.00E+00	0.00E+00	0.00E+00	2.87E-03	9.38E-01	-3.86E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.74E-03	5.69E-03	6.45E-08	3.17E+00
	Best	2.19E-224	2.36E-115	3.76E-219	1.63E-90	0.00E+00	0.00E+00	2.00E+01	1.00E+01	3.98E-01	-3.86E+00

Table 8. Comparison of model prediction performance.

Classifier	Precision	Recall	F1-Score	Accuracy	RMSE	MAE
RF	0.74	0.86	0.79	0.7471	0.5029	0.2529
GWO-RF	0.78	0.78	0.77	0.7816	0.3866	0.3122
GA-RF	0.78	0.78	0.77	0.7816	0.3844	0.3063
DE-RF	0.77	0.77	0.76	0.7701	0.3989	0.3294
POA-RF	0.73	0.72	0.72	0.7126	0.4776	0.3037
IPOA-RF	0.85	0.86	0.86	0.8506	0.3148	0.2122

is relatively sensitive to the variations in θ and k, while the influence of v remains comparatively stable. Specifically, when θ is within the range [0.5,0.6] and k is within [1.6,2.0], the convergence performance of the algorithm is optimal, demonstrating that the proposed parameter design is both reasonable and stable.

4.4 Analysis of IPOA-RF Model Prediction Results

The 431 groups of slope data were divided into training and testing sets in a ratio of 8:2. As shown in Figure 12, the prediction accuracy of the IPOA-RF model reached 85.1%, which is about 10.4% higher than that of the random forest model with an accuracy of 74.7%. This demonstrates that the slope stability prediction based on the IPOA-RF model is scientifically reasonable and reliable.

4.5 Evaluation of IPOA-RF Model Prediction Performance

To evaluate the effectiveness of the proposed IPOA-RF in slope stability classification, this study conducted training and testing on the same dataset and compared the IPOA-RF model with the original RF model and four hybrid optimization models (GWO-RF, GA-RF, DE-RF, and POA-RF). The evaluation metrics included Precision, Recall, F1-Score, and Accuracy, as well as RMSE and MAE based on probability output. Among them, the classification metrics were calculated using weighted averages to reduce the influence of class imbalance, while RMSE and MAE measured the continuous deviation between the predicted probabilities and the actual binary labels.

On the slope dataset used in this study, the six performance metrics of IPOA-RF reached 0.85, 0.86, 0.86, 0.8506, 0.3148, and 0.2122, respectively. Compared with the unoptimized random forest, the improved model demonstrates higher overall classification and prediction capability. Consistent with other swarm intelligence–based hybrid models, the IPOA-RF model achieved the best or near-best performance across all metrics (see Table 8).

5 Conclusion

This study proposed a slope stability prediction model (IPOA-RF) based on an improved Pelican Optimization Algorithm combined with a Random Forest classifier. By introducing logistic chaotic mapping, reverse learning, and adaptive distribution perturbation mechanisms into the traditional POA, the global exploration and local exploitation abilities of the algorithm were effectively enhanced, thereby improving the efficiency and accuracy of Random Forest hyperparameter optimization.

Experimental results show that the proposed IPOA-RF model outperforms four other hybrid optimization-based Random Forest models (GWO-RF, POA-RF, GA-RF, and DE-RF) and six conventional machine learning models (SVM, RF, KNN, XGBoost, CNN, and LSTM). In terms of evaluation metrics such as Accuracy, Precision, Recall, F1-Score, RMSE, and MAE, the IPOA-RF model consistently achieved the best results, confirming the superior performance and reliability of the proposed method in slope stability prediction.

This study has the following practical implications for slope engineering:

1. It improves the accuracy of slope stability prediction, providing a more reliable auxiliary tool for engineering practice.
2. It enables the construction of a quantitative evaluation framework for slope stability, offering scientific support for design optimization and risk reduction.
3. It facilitates the early identification of potential failure risks, thereby supporting the formulation of preventive and mitigation measures.
4. Integrating the prediction results into engineering management can enhance both the safety and economic efficiency of slope engineering.

Nevertheless, this study has certain limitations. First,

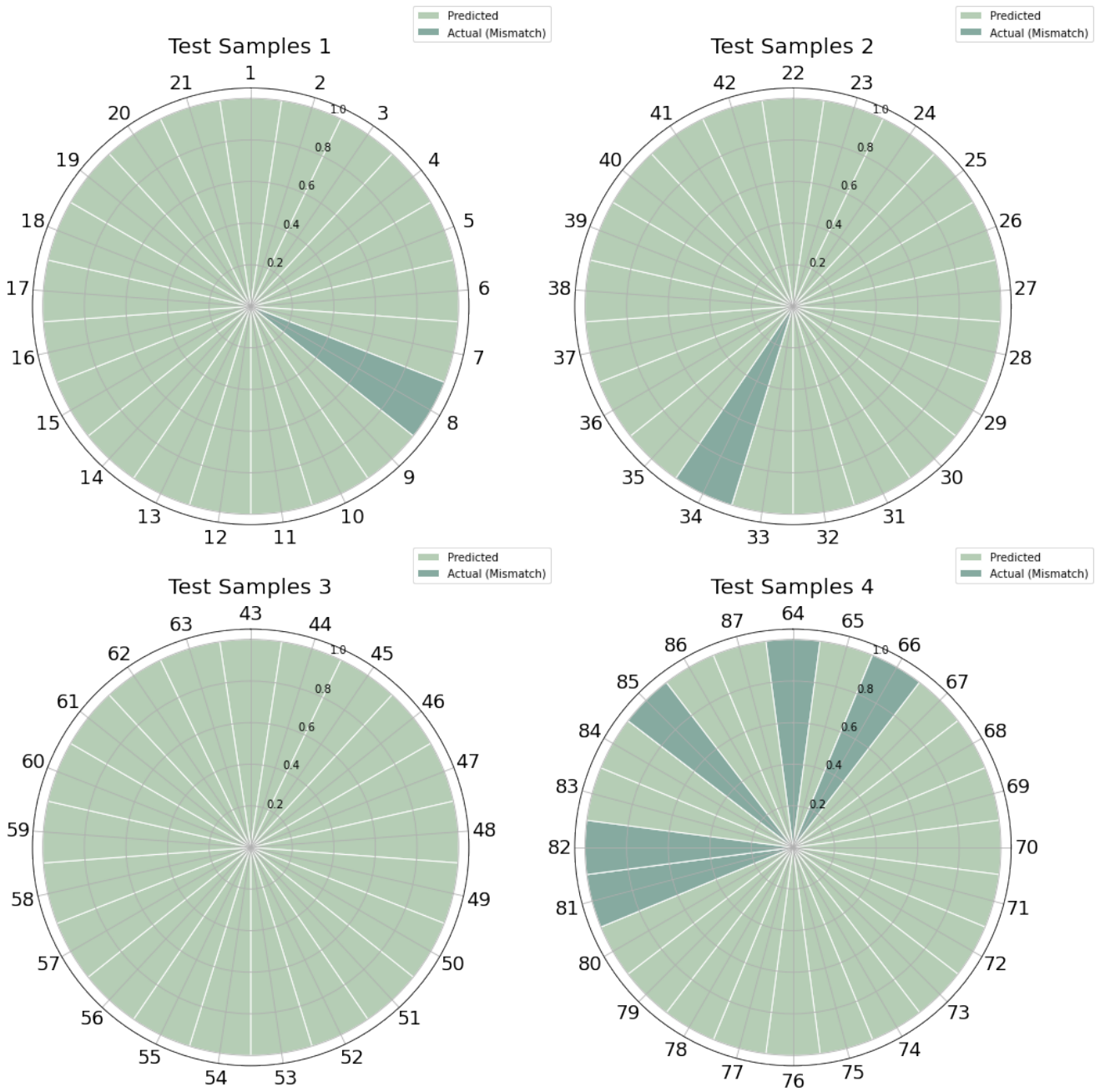


Figure 12. Prediction results of the IPOA-RF model.

the experimental dataset includes only 431 groups of slope cases, and the limited sample size may affect the generalizability of the model. Second, the improvement in prediction performance is, to some extent, achieved at the expense of reduced model interpretability, which is a common issue in machine learning models. Finally, this paper focuses solely on slope stability as a classification problem; future research will integrate multi-factor safety data for regression and risk analysis to provide a more comprehensive perspective. Future work will primarily focus on expanding the dataset scale, optimizing the selection of prediction indicators and sensitivity analysis, and further exploring the IPOA-RF method in other geotechnical engineering applications to validate its generalizability and engineering value.

Data Availability Statement

Data will be made available on request.

Funding

Jiandong Zhang acknowledges the support of the National Natural Science Foundation of China under Grant 12526628, the Youth Science and Technology Talent Special Fund Project of Lanzhou Science and

Technology Bureau under Grant 2025-QN-101, the Young Doctor Fund Project of Gansu Provincial Department of Education under Grant 2026QB-016, the College Teachers Innovation Foundation Project of Gansu Provincial Department of Education under Grant 2024A-002, and the Innovative Fundamental Research Group Project of Gansu Province under Grant 25JRRA002.
Rongfang Yan is partially supported by the National Natural Science Foundation of China under Grant 12361060 and the Industrial Support Fund Program of Gansu Provincial Department of Education under Grant 2025CYZC-016.
Huanhuan Zhao, Xiaole Zhao, and Xueyi Wen jointly acknowledge the support of the Northwest Normal University Postgraduate Research Funding Project under Grant KYZZ2025-LXS101.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Ullah, S., Khan, M. U., & Rehman, G. (2020). A brief review of the slope stability analysis methods. *Geol. Behav, 4*(2), 73-77. [CrossRef]

[2] Azarafza, M., Akgün, H., Ghazifard, A., Asghari-Kaljahi, E., Rahnamarad, J., & Derakhshani, R. (2021). Discontinuous rock slope stability analysis by limit equilibrium approaches–a review. *International Journal of Digital Earth, 14*(12), 1918-1941. [CrossRef]

[3] Sengani, F., & Mulenga, F. (2020). Application of limit equilibrium analysis and numerical modeling in a case of slope instability. *Sustainability, 12*(21), 8870. [CrossRef]

[4] Ureel, S., & Momayez, M. (2014). An investigation of the limit equilibrium method and numerical modeling for rock slope stability analysis. In *Rock mechanics and its applications in civil, mining, and petroleum engineering* (pp. 218-227). [CrossRef]

[5] Li, L. C., Tang, C. A., Zhu, W. C., & Liang, Z. Z. (2009). Numerical analysis of slope stability based on the gravity increase method. *Computers and Geotechnics, 36*(7), 1246-1258. [CrossRef]

[6] Gordan, B., Jahed Armaghani, D., Hajihassani, M., & Monjezi, M. (2016). Prediction of seismic slope stability through combination of particle swarm optimization and neural network. *Engineering with computers, 32*(1), 85-97. [CrossRef]

[7] Chakraborty, A., & Goswami, D. (2017). Prediction of slope stability using multiple linear regression (MLR) and artificial neural network (ANN). *Arabian Journal of Geosciences, 10*(17), 385. [CrossRef]

[8] Kasa, A., & Mohd, S. F. (2024). Performance Prediction Evaluation of Machine Learning Models for Slope Stability Analysis: A Comparison Between ANN, ANN-ICA and ANFIS. *J. Electr. Syst, 20*, 4364-4374.

[9] Huang, S., Huang, M., & Lyu, Y. (2020). An improved KNN-based slope stability prediction model. *Advances in Civil Engineering, 2020*(1), 8894109. [CrossRef]

[10] Zhang, W., Li, H., Han, L., Chen, L., & Wang, L. (2022). Slope stability prediction using ensemble learning techniques: A case study in Yunyang County, Chongqing, China. *Journal of Rock Mechanics and Geotechnical Engineering, 14*(4), 1089-1099. [CrossRef]

[11] Demir, S., & Sahin, E. K. (2023). Random forest importance-based feature ranking and subset selection for slope stability assessment using the ranger implementation. *Avrupa Bilim ve Teknoloji Dergisi*, (48), 23-28. [CrossRef]

[12] Kurnaz, T. F., Erden, C., Dağdeviren, U., Demir, A. S., & Kökçam, A. H. (2024). Comparison of machine learning algorithms for slope stability prediction using an automated machine learning approach. *Natural Hazards, 120*(8), 6991-7014. [CrossRef]

[13] Huang, F., Xiong, H., Chen, S., Lv, Z., Huang, J., Chang, Z., & Catani, F. (2023). Slope stability prediction based on a long short-term memory neural network: comparisons with convolutional neural networks, support vector machines and random forest models. *International Journal of Coal Science & Technology, 10*(1), 18. [CrossRef]

[14] Rukhaiyar, S., Alam, M. N., & Samadhiya, N. K. (2018). A PSO-ANN hybrid model for predicting factor of safety of slope. *International Journal of Geotechnical Engineering, 12*(6), 556-566. [CrossRef]

[15] Zhou, J., Li, E., Yang, S., Wang, M., Shi, X., Yao, S., & Mitri, H. S. (2019). Slope stability prediction for circular mode failure using gradient boosting machine approach based on an updated database of case histories. *Safety Science, 118*, 505-518. [CrossRef]

[16] Li, M., Li, K., Qin, Q., & Yue, R. (2023). Slope stability prediction based on IPOARF algorithm: A case study of Lala Copper Mine, Sichuan, China. *Expert Systems with Applications, 229*, 120595. [CrossRef]

[17] Xie, H., Dong, J., Deng, Y., & Dai, Y. (2022). Prediction model of the slope angle of rocky slope stability based on random forest algorithm. *Mathematical problems in engineering, 2022*(1), 9441411. [CrossRef]

[18] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July).

A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152). [CrossRef]

[19] Yang, Y., Zhou, W., Jiskani, I. M., Lu, X., Wang, Z., & Luan, B. (2023). Slope stability prediction method based on intelligent optimization and machine learning algorithms. *Sustainability, 15*(2), 1169. [CrossRef]

[20] Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors, 22*(3), 855. [CrossRef]

[21] Csillik, O., Evans, I. S., & Drăguţ, L. (2015). Transformation (normalization) of slope gradient and surface curvatures, automated for statistical analyses from DEMs. *Geomorphology, 232*, 65-77. [CrossRef]

[22] Pan, C. (2025). Novel and hybrid model of pelican optimization algorithm with light gradient boosting model for smart infrastructure: A high-precision approach to occupancy and failure prediction. *Green Technologies and Sustainability*, 100323. [CrossRef]

[23] Khajehzadeh, M., & Keawsawasvong, S. (2023). Predicting slope safety using an optimized machine learning model. *Heliyon, 9*(12), e23012. [CrossRef]

[24] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297. [CrossRef]

[25] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). [CrossRef]

Huanhuan Zhao is currently pursuing an Master degree in Applied Statistics at the College of Mathematics and Statistics, Northwest Normal University. Her research focuses on the statistical machine learning methods of system safety and slope stability. (Email: sunny_zhao.fz@foxmail.com)

Xiaole Zhao is currently pursuing an Master degree in Applied Statistics at the College of Mathematics and Statistics, Northwest Normal University. His research focuses on the application of statistical modeling, machine learning methods and prediction. (Email: 1942957695@qq.com)

Xueyi Wen is currently pursuing an Master degree in Applied Statistics at the College of Mathematics and Statistics, Northwest Normal University. Her research focuses on the application of statistical machine learning theory in system stability and safety. (Email: wen15339078129@163.com)

Rongfang Yan is currently a professor from Northwest Normal University. His research interests include reliability, mathematical statistics, reliability theory, statistical dependence. (Email: yanrf@nwnu.edu.cn)

Jiandong Zhang is currently an associate professor from Northwest Normal University. His research interests include reliability theory and dependent risk models. (Email: jdzhang@nwnu.edu.cn)