Analysis of Slope Stability using Innovative Hybrid BPSO-SVM Machine Learning Techniques for Enhance Environmental Sustainability

Saurabh Kumar Anuragi[1], D. Kishan[2]

[1]*Ph.D Scholar, Department of Civil Engineering, Maulana Azad National Institute of Technology, Bhopal 462003, India*
[2]*Associate Professor, Department of Civil Engineering, Maulana Azad National Institute of Technology, Bhopal 462003, India*

Abstract: This study aims to improve the forecasting performance of slope stability for impacting environmental sustainability and infrastructure safety predictions by using the Binary Particle Swarm Optimization (BPSO) coupled with Support Vector Machine (BPSO-SVM) models. The BPSO technique is utilized to select relevant features from the dataset, thereby improving the overall effectiveness of the predictive models. The research includes 108 slope stability examples, with the dataset split between 70% training and 30% validation. The dataset comprises seven input parameters: cohesiveness, slope angle, unit weight, angle of internal friction, slope height, pore water pressure coefficient, and factor of safety. The objective is to classify the slope status, turning the problem into a classification task. To obtain optimal hyper-parameters for the SVM model, Grid Search was exploited. The accuracy of the slope stability predictions given by several models was assessed using receiver operating characteristic (ROC) curves. The results indicate that the BPSO-SVM model outperforms the standalone SVM and BPSO models, serving as a robust computational tool capable of accurately predicting slope stability to enhance the environmental sustainability.

Keywords: Slope Stability, Environmental Sustainability, Soil Quality Assessment, Support Vector Machine, Binary Pso, Grid Search, Cross Validation,Binary Particle Swarm Optimization(BPSO).

1. Introduction

Slope stability analysis is a fundamental aspect of geotechnical engineering, playing a crucial role in ensuring the safety and sustainability of infrastructure projects, such as highways, dams, and open-pit mines. Slope failures can lead to significant economic losses, environmental degradation, and human casualties, making accurate stability prediction essential for disaster prevention and risk management. Predicting slope stability accurately is a significant challenge due to the complex nature of soil's physical properties. The increase in slope failures, which lead to substantial economic and environment losses, has caught the attention of researchers and engineers. It is crucial to analyze and stabilize slopes in order to mitigate or prevent such damages, requiring a thorough understanding of the underlying mechanisms responsible for slope failures. Nevertheless, extensive efforts have been dedicated to minimizing losses through the use of numerical and analytical modelling techniques, which enable precise predictions and facilitate appropriate actions [1]. The finite element approach and the limit equilibrium method are the two most used methods for studying slope stability. These strategies are critical for assessing slope stability and making choices that promote safety while mitigating hazards [2–10]. However, they depend on linear connection models, which may not account for non-linear deformations. The sliding surface is assumed to achieve the final state of failure at the same time, regardless of the slip surface's real stress conditions [11–15]. To overcome this constraint, numerous studies have merged the strength reduction approach with finite element analysis for slope stability analysis [16–18]. This integrated method seeks to calculate the factor of safety by including a failure condition to assess if the system achieves limit equilibrium. As a consequence, doing slope stability analysis using traditional approaches requires a thorough knowledge, sophisticated modeling techniques, and a large quantity of experimental data.

In current years, intelligent computational techniques have been widely used in geotechnical applications such as landslide susceptibility assessment [19–22], foundation analysis [23–27], predicting pile foundation bearing capacity [28, 29], predicting soil physical properties [30–32], soil quality assessment [33, 34], and liquefaction assessment [35–37]. Meanwhile, amazing results were achieved by expanding the available slope parameters for slope stability prediction using supervised learning approaches . Numerous studies have documented the use of support vector machines (SVM) in geotechnical engineering and shown that it outperformed other machine learning approaches [38–41]. Similarly, Particle Swarm Optimization (PSO) has been effectively used to a variety of geotechnical engineering challenges. Both of these modeling techniques need just a few user-defined parameters and produce global minima [42–46]. However, PSO has flaws such as premature convergence and a lack of variety, which limit its capacity to identify global optima. It is sensitive to parameter adjustments and has difficulty with limited tasks. PSO lacks problem-specific information and scalability in high-dimensional domains, however upgrades and adjustments have been suggested to address these constraints. Furthermore, it has been shown that the support vector machine (SVM) may be used as part of a hybrid model with other machine learning approaches to reliably forecast slope stability. For example, Xue et al. [47] used PSO-SVM and GS-SVM hybrid models to enhance slope stability prediction by determining the appropriate SVM parameters. The findings confirmed the usefulness of PSO-SVM in this setting. Similarly, Xinhua Xu. [48] effectively used a hybrid model integrating least square support vector machine (LSSVM) and particle swarm optimization (PSO) to improve the forecast capabilities of slope stability analysis. Other techniques, such as using cloud models [49], Monte Carlo simulation [50], and comparing SVM to other models like ANN, have further shown SVM's promise in slope stability prediction. Furthermore, changes to algorithms such as the gravitational search algorithm (GSA) have resulted in better performance. Despite these advances, it is critical to choose an appropriate approach for slope deformation prediction, since each of the methods discussed above has limits and downsides.

As a consequence, other methodologies are required for more accurate slope stability forecasts. Taking into account the superior performance of SVM and the constraints of PSO, an attempt has been made to study the possible usage of an improved SVM with Binary Particle Swarm Optimization (BPSO) in building a prediction function for determining slope status using available parameters. In addition, a comparison of SVM and BPSO-SVM is performed to evaluate the models' prediction capabilities. The SVM is a machine learning methodology developed by Vapnik [51] that is well-known for its ability to generate statistical predictions and categorization. It has been popular among academics owing to its user-friendliness, solid features, and great performance. While SVM has been extensively utilized in several disciplines, including neurology, bioinformatics, finance, and meteorology. However, there is still need for further research into its potential uses in civil engineering outside geotechnical engineering. Despite some successful efforts, SVM outperformed the Artificial Neural Network (ANN) method, indicating that it is a viable technology for civil engineering applications. As a result, further study is needed to uncover the potential of SVM in civil engineering and investigate its capabilities in the sector.

2. Support Vector Machines

An inner product function defines the nonlinear transformation used by the Support Vector Classifier (SVC), a machine learning algorithm that maps the input space into a higher-dimensional feature space. The theoretical underpinnings of SVC are based on linear separability, decision boundaries, and margin maximization. Vapnik and Chervonenkis (1963) [52] developed the concept of the Vapnik-Chervonenkis (VC) dimension, which provided the foundation for understanding SVC's generalization performance. The VC dimension measures the capacity or complexity of a hypothesis space, which is the set of decision boundaries that a learning algorithm may learn. In the context of SVC, the VC dimension reflects the greatest number of points that may be fractured or completely separated by the algorithm's learnt decision boundary. comprehension SVC's generalization performance requires a comprehension of the VC dimension. It offers a theoretical basis for examining the trade-off between model complexity and the capacity to generalize successfully to new data.

Boser et al. (1992) [53] introduced kernel functions, which enhanced SVC's capacity to handle non-linearly separable data. Cortes and Vapnik (1995) [51] developed the Support Vector Classifier (SVC) formulation, demonstrating its capacity to discover optimum separating hyperplanes with greatest margins by incorporating two terms, namely slack variable ξ and penalty factor C. The slack variable ξ represents the standard deviation of a data pattern from the ideal state, while the penalty factor C determines the trade-off between the amount of misclassifications in training data.

The decision functions for different conditions are :

For Linearly Separable :

$$y_i[(w^T x_i) + b] - 1 \geq 0 \quad (1)$$

For Linearly Inseparable :

$$y_i[(w^T x_i) + b] \geq 1 - \xi_i \quad (2)$$

To minimize,

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \quad (3)$$

For Non-linear Classification :

Linear Kernel :

$$k(x_i x_j) = x_i^T x_j \quad (4)$$

Polynomial Kernel :

$$k(x_i x_j) = (\Upsilon x_i^T + r)^d, \Upsilon > 0 \quad (5)$$

Radial Basis Function (RBF) :

$$k(x_i x_j) = e^{(\Upsilon ||x_i - x_j||^2)}, \Upsilon > 0 \quad (6)$$

Sigmoid Kernel :

$$k(x_i x_j) = \tan (\Upsilon x_i^T x_j + r) \quad (7)$$

Where w is an adaptive weight factor, x is an input vector, b is bias and w^T x is an inner product of w and x and ϒ, r and d are kernel parameters.

Numerous research have focused on improving SVC training and performance. Platt (1999) [54] presented a sequential minimum optimization approach to increase the efficiency of training large-scale datasets. Joachims (2006) [55] proposed the Budgeted Support Vector Machine, which reduced training time by picking a subset of support vectors. Additional research has looked at the use of parallel computing, distributed learning, and active learning approaches to speed up training and enhance scalability. From its theoretical roots to its implementations in a variety of fields, SVC has shown exceptional performance and adaptability. While issues like as parameter tuning and scalability persist, continuing research seeks to solve these limits and improve the algorithm's performance. Understanding SVC's advances and future prospects allows academics to contribute to its continuous development and explore its possibilities in tackling complicated classification challenges.

3. Binary PSO

Binary Particle Swarm Optimization (BPSO) is an optimization technique that solves binary optimization issues. In binary optimization issues, the aim is to determine which binary string optimizes or minimizes a particular objective function. Binary PSO is a heuristic approach to simulating the behavior of a swarm of particles in a multidimensional search space. In Binary PSO, particles represent locations in a binary space, with each element of a particle's position vector having only the values 0 or 1. In other words, $x_i \in B^{n_x}$ or x_{ij} can only be 1 or 0. The location of a particle is updated by flipping one or more bits in the particle's binary string representation. This essentially moves the particle around various corners of a hypercube in the binary search space. The flipping of bits might cause the particle to move closer or further away from the ideal solution.

The binary PSO method starts by randomly creating a population of particles, then initializes their locations and velocities. The fitness function is then computed for each particle, and the optimal placements of both individual particles and the whole swarm are changed appropriately. The swarm best location is the position of the particle with the highest fitness function value throughout the whole population. Next, the velocity and location of each particle are updated using the following equation:

$$V_{[i,j]} = w * V_{[i,j]} + c1 * \text{rand } 1 * \left(p_{best[i,j]} - x_{[i,j]}\right) + c2 * \text{rand2} * \left(g_{best[i,j]} - x_{[i,j]}\right) \quad (8)$$

$$x_{[i,j]} = 1, \text{if rand}(0) < \text{sigmoid } (V_{[i,j]})$$
$$x_{[i,j]} = 0, \text{otherwise}$$

where $V_{[i,j]}$ is the velocity of the j^{th} dimension of the i^{th} particle, w is the inertia weight, c1 and c2 are the cognitive and social learning factors, rand1 and rand2 are random numbers between 0 and 1, $p_{best[i,j]}$ is the best position of the i^{th} particle in the j^{th} dimension, and $g_{best[j]}$ is the best position of the swarm in the j^{th} dimension. The sigmoid function is used to convert the velocity into a probability of flipping the bit from 0 to 1.

The BPSO defines velocities and particle trajectories based on whether each bit is set to 1 or 0. For example, a velocity V_{ij} (t) of 0.3 means that the corresponding bit has a 30% probability of being set to 1 and a 70% chance of being set to 0. To guarantee that velocities are perceived as probabilities, they are usually limited to the range [0, 1]. There are many techniques for normalizing velocities such that V_{ij} is between 0 and 1. One popular method is to divide each V_{ij} by the greatest velocity, $V_{max.j}$. If $V_{max,j}$ is big and the real velocity V_{ij} (t) << $V_{max,j}$ for all time steps, the range of velocities decreases, reducing the probability of a position changing to bit 1. For example, if $V_{max,j}$ = 10 and V_{ij} (t) = 5, the normalized velocity V'_{ij} (t) is 0.5, implying a 50% likelihood that x_{ij} (t + 1) = 1. Using this normalization strategy might cause early convergence to inferior solutions since it restricts the algorithm's exploration abilities.

The velocity normalization is obtained by using sigmoid function,

$$V_{ij}(t) = \text{sig}\left(V_{ij}(t)\right) = \frac{1}{1+e^{-V_{ij}(t)}} \quad (9)$$

Using (9), the position update changes to

$$x_{ij}(t+1) = \begin{cases} 1, & \text{if } r_{ij}(t) < \text{sig}(V_{ij}(t+1)) \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

Where r_{ij} is a uniform random number in the range [0,1]. The more detailed expressions can be referred at [56].

4. Analysis Data Visualization and Preprocessing

The major goal of the presented work is to improve the performance of the SVM model by optimizing its parameters for a given dataset. Furthermore, a comparison is made between SVM and BPSO-SVM. The dataset used in this investigation comes from [57]. The dataset includes unit weight (ϒ), cohesion (c), angle of internal friction (ϕ), slope angle (β), height (H), pore water pressure coefficient (ru), factor of safety (FS), and slope status (S) (stable or unstable). The FS is a comprehensive instrument for analyzing the stability of a slope. It entails computing the slope's sliding resistance to its sliding force, which is directly proportional to the soil's shear strength. In this research work, the slope status is represented as a binary value, with '1' indicating stable and '0' unstable. The present study uses BPSO for feature optimization and SVM for predicting status. The Table 1 provides statistical insights, such as standard deviation, minimum, mean, maximum, average, variance, and count. These metrics help to understand the variation, central tendencies, and sample size of the data points within the study.

Table 1 Description of Dataset

index	ϒ (kN/m^3)	C (kPa)	ϕ (^{0})	β (^{0})	H (m)	ru
Standard deviation	3.68	22.40	11.25	9.67	47.37	0.17
min	12.00	0.00	0.00	16.00	3.60	0.00
mean	20.01	14.58	26.25	33.44	44.28	0.19
max	28.44	150.05	45.00	53.00	214.00	0.50
Average	19.96	10.48	26.63	33.26	40.73	0.21
Variance	12.53	116.73	116.57	90.23	1845.94	0.03
count	108.00	108.00	108.00	108.00	108.00	108.00

The Violin plots of the dataset are given in Fig. 1. The violin plot shows the distribution and density of a dataset across categories or groupings. The breadth of the violin at each position represents the density of data at that location. The violin's thicker sections symbolize high-density zones, while the thinner parts represent low-density places. The horizontal line within the violin reflects the data's median value.

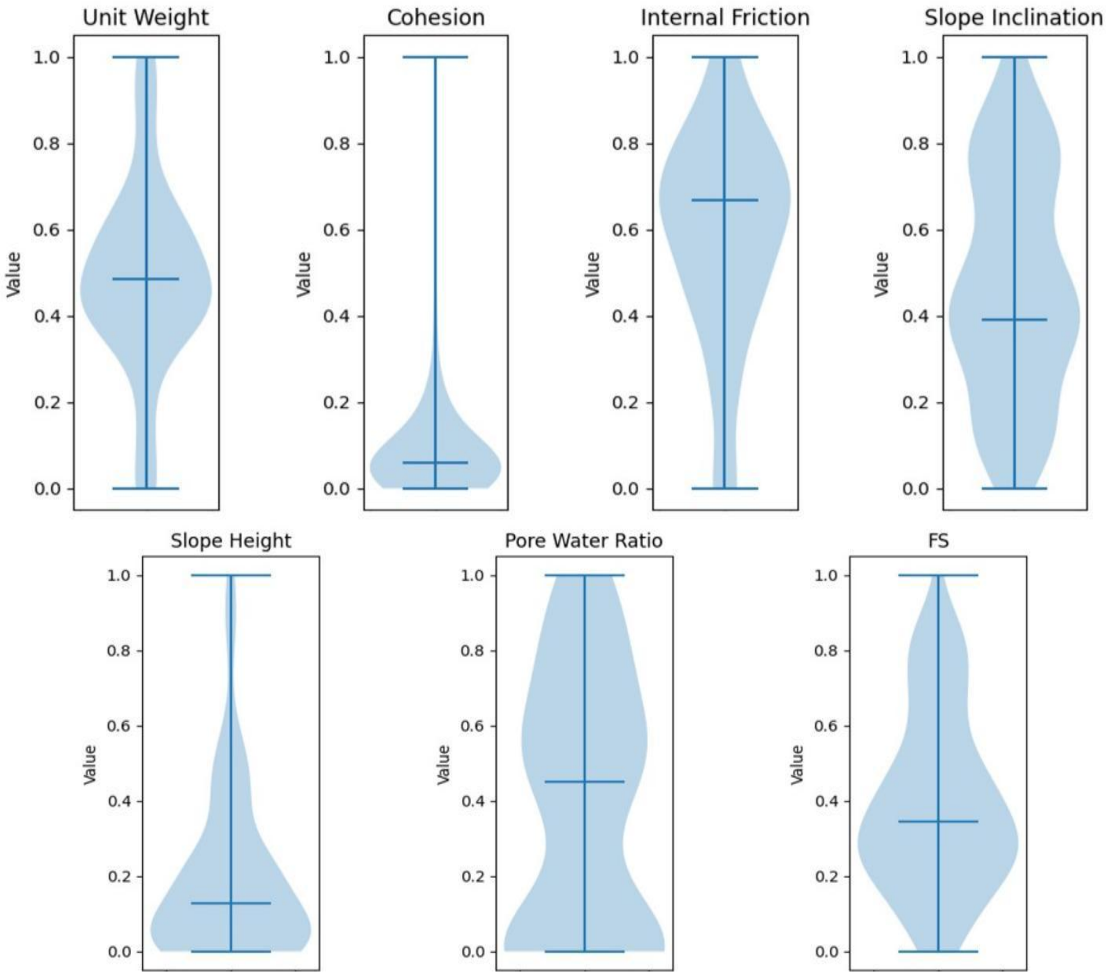

Fig. 1. Violin Plots showing distribution of slope cases

The SVM model's hyper-parameters are calculated using Grid Search CV, whilst the BPSO-SVM hyper-parameters are achieved by trial-and-error to attain the best AUC, as shown in Tables 2 and 3, respectively.

Table 2 SVM Parameters Utilized for Grid Search

Kernels	Parameters Range
RBF	C = [1,10,20,30,40,50,60,70,80,90,100] ; gamma = [0.01,0.1,1,10]
Poly	C = [1,10,20,30,40,50,60,70,80,90,100] ; degree = [1,2,3,4,5,6]
Linear	C = [1,10,20,30,40,50,60,70,80,90,100]

Table 2 summarizes the different kernel types and parameter ranges employed in a machine learning experiment. Kernels are critical components of machine learning algorithms, especially Support Vector Machines (SVMs), which are used for classification tasks. Different kernel functions influence the decision boundary and hence the model's performance. For complete investigation, the parameter values for several kernel functions are selected at random. Table 3 show the set of parameters used in the BPSO method. In this study, we used acceleration coefficients c1=c2=2, 50 particles, 500 iterations, 0.9 inertia weight, and alpha values ranging from 0.1 to 0.9.

Table 3 Utilized Parameters in the BPSO Model

Parameters	Values
Acceleration coefficients (c1,c2)	[2,2]
Inertia Weight (w)	0.9
Number of dimensions (k)	7
Number of particles (n_particles)	50
Iterations (iter)	500
alpha	[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]

The aforementioned seven parameters (ϒ,c,ϕ,β,H,ru,FS) are used as input parameters while status is the output for the model. Before the datasets were trained, each parameter of the dataset is normalized between 0 and 1 using (11).

$$y_{normalization} = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (11)$$

Where, y is a normalized input parameter, x is the original input parameter, and x_{max} and x_{min} are the maximum and minimum values, respectively. In SVM classification, the dataset is split 70:30, with 70% for training and 30% for testing. Following that, both SVM and BPSO-SVM models undergo extensive hyper-parameter adjustment. Following that, models are built using the found optimum hyper-parameters for both the SVM and BPSO-SVM techniques. These models are validated using the Receiver Operating Characteristic (ROC) curve, which is an important method for measuring and comparing model performance. Fig. 2 shows a flow chart of the process.

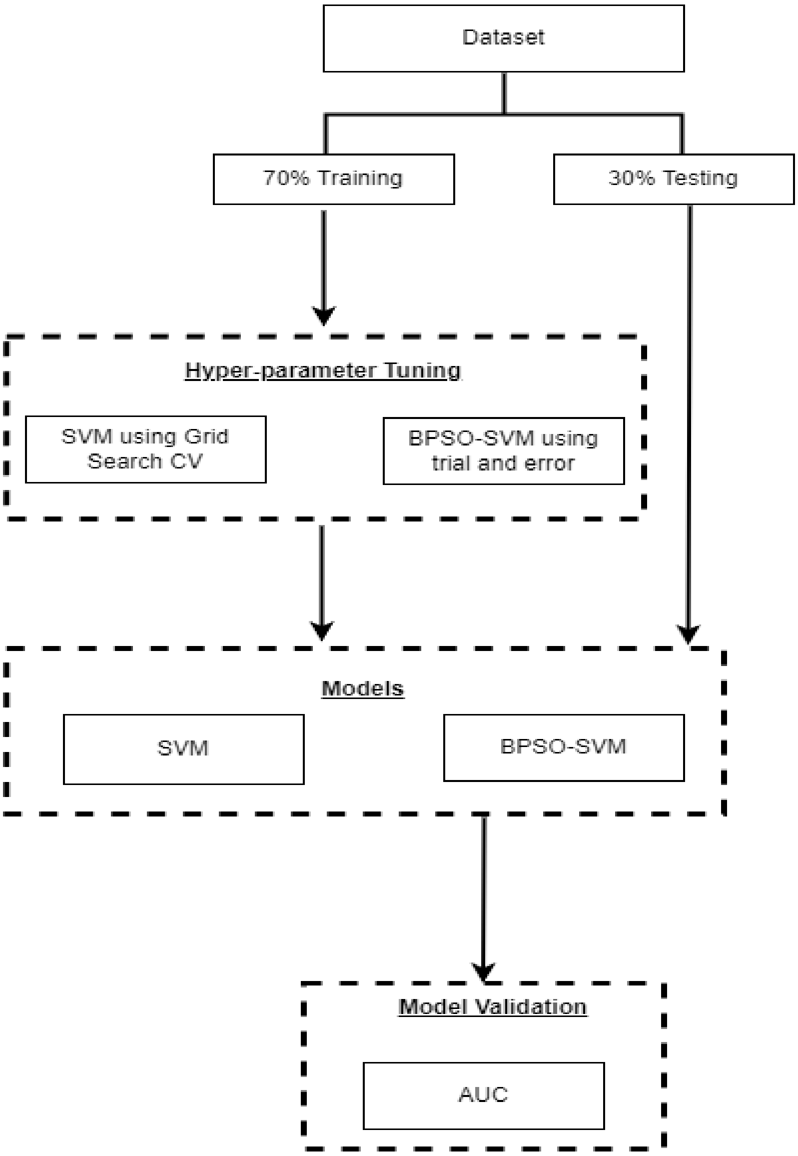

Fig. 2. Methodology of the Study

Fig. 3 shows the sequential process of BPSO-SVM, which was built with the goal of improving SVM's effectiveness. This augmentation is accomplished by simultaneously completing two tasks: choosing relevant feature subsets from a dataset and improving SVM parameters. Each particle in the BPSO component represents a possible feature subset. The particle's positional vector, which operates in the binary domain (0 or 1), determines whether a given characteristic is present (1) or absent (0). Each subset's classification accuracy is evaluated using a fitness function. Notably, the fitness function that yields better accuracy is used to evaluate solutions and update particle locations. This iterative technique helps to uncover feature subsets that make a substantial contribution to accurate categorization. SVM is used after picking the best-performing subgroup. The improved parameters from BPSO are then used to train and evaluate the data using the tuned SVM to get better classification results.

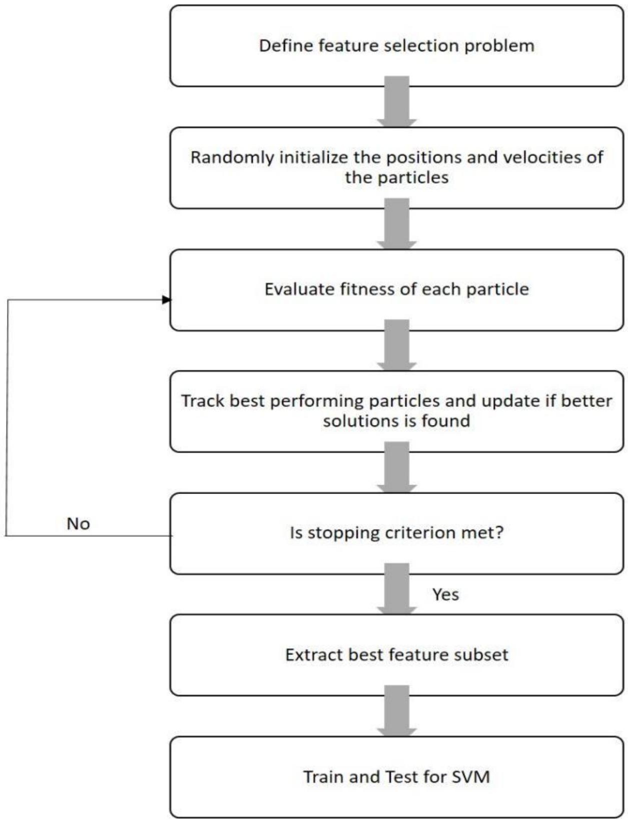

Fig. 3. Flow Chart of the BPSO-SVM model

5. Results and Discussion

Initially, SVM is applied to the dataset using several kernel functions, including linear, polynomial, and rbf. Table 4 shows the best hyper-parameters for each SVM kernel as derived using Grid Search Cross-Validation (CV). Given the Grid Search CV score, the best hyper-parameters for the SVM-rbf kernel are C = 20 and gamma = 0.1, with a CV score of 0.96. For SVM-poly, the best settings are C = 1 and degree = 1, with a CV score of 0.946. Finally, for SVM-linear, the optimal hyper-parameter is C = 10, which corresponds to a CV score of 0.92.

Table 4 Optimal Hyperparameters Obtained for SVM using Grid Search

Kernels	Optimal Hyperparameters	Grid Search CV Score
RBF	C = 20, gamma = 0.1	0.96
Poly	C = 1, Degree = 1	0.946
Linear	C = 10	0.92

Subsequently, We then used the Binary Particle Swarm Optimization (BPSO) technique to optimize the dataset parameters. This entailed a trial-and-error procedure for fine-tuning the BPSO hyper-parameters to improve the SVM model's performance. The primary goal of the BPSO was to attain the maximum AUC value for the BPSO-SVM model at various alpha levels. Table 5 shows the ideal alpha value that resulted in the lowest cost for each kernel function of SVM. The Table 5 also includes information on the dataset's chosen features for the best SVM hyperparameters.

Table 5 Features Selected by BPSO Model for Optimal Alpha Values with Different SVM Kernels

Kernels	alpha	Best Cost	Selected Features
RBF	0.9	0.0429	ϒ,c,ru,FS
Poly	0.9	0.0286	ϒ,c,H,ru,FS
Linear	0.9	0.0433	ϒ,c,ϕ,H,ru,FS

The cost in the BPSO framework relates to a numerical value supplied to the fitness function that is presently being optimized. The basic goal of BPSO is to reduce costs, which entails identifying the solution that offers the best feasible result based on the issue criteria. BPSO used four features (ϒ,c,ru,FS) from a dataset to get the optimal cost of 0.0429 for the SVM-rbf model. Similarly, SVM-linear finds six features (ϒ,c,ϕ,H,ru,FS) with the best possible cost of 0.0433. In SVM-poly, a collection of five features (ϒ,c,H,ru,FS) results in a better outcome with a best cost of 0.0286. The selection of features, which leads to their respective optimum costs, is consistently accomplished with an alpha value of 0.9. Fig. 4 depicts the oscillations in AUC for the BPSO-SVM model across a variety of alpha values. Notably, SVM-rbf and SVM-poly achieve equal results (AUC = 1.00), outperforming SVM-linear (AUC = 0.969) at an ideal alpha value of 0.9.

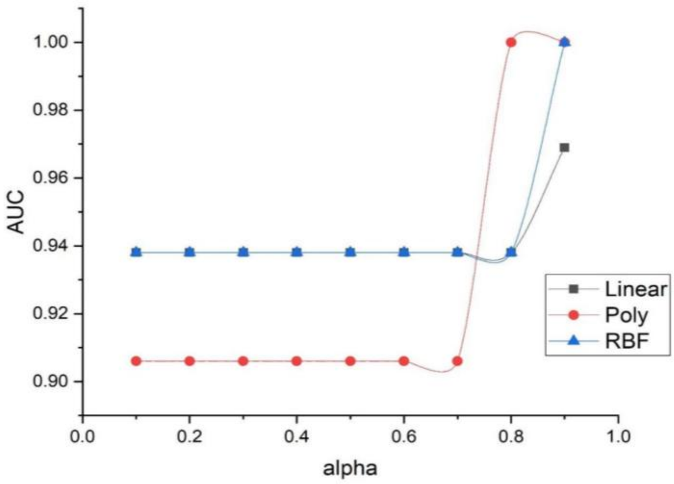

Fig. 4. AUC Variations of the BPSO-SVM Model Across Different Alpha Values

The performance comparison of SVM and BPSO-SVM models is illustrated in Fig. 5 through ROC curves. These graphs represent the trade-off between the true positive rate and the false positive rate, demonstrating the BPSO-SVM model's improved performance in terms of classification accuracy and discriminating capabilities over the traditional SVM technique. The findings in Fig. 5a reveal that both the rbf and linear kernels performed equally in terms of AUC (SVM-rbf = 0.938 and SVM-linear = 0.938), demonstrating their equivalent discriminative powers. Both kernels outperformed the poly kernel (SMV-poly = 0.906), suggesting that the poly kernel may have limited ability to capture the dataset's complexity. However, Fig. 5b shows that the rbf and poly kernels displayed comparable levels of efficacy (SVM-rbf = 1.00 and AUCSVM-poly = 1.00), outperforming the linear kernel (SMV-linear = 0.969) not only in terms of AUC but also in best cost, as shown in Table 5. This suggests that the rbf and poly kernels were better at generating higher AUC values while reducing the cost function, making them good options for the SVM model.

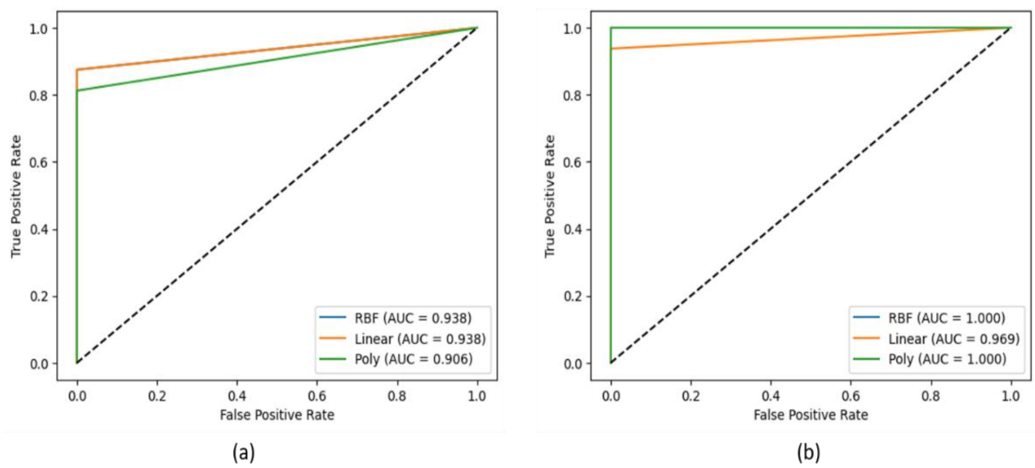

Fig. 5. Comparing ROC Curves for Different Kernel Functions: (a) SVM with Optimal Hyperparameters, and (b) BPSO-SVM Model with Feature Optimization

It is observed that there is a significant improvement in the performance of the BPSO-SVM model when utilizing the 'Poly' kernel. This improvement was demonstrated by an AUC of 1.00 and a best cost value of 0.0286. These results highlight the efficacy of the BPSO optimization approach in enhancing the model's predictive capabilities.

6. Conclusion

This work proposes a hybrid model that combines Support Vector Machine (SVM) and Binary Particle Swarm Optimization (BPSO) to improve predicting performance. The BPSO-SVM model combines these two revolutionary approaches, with SVM focusing on structural minimization rather than error reduction, and BPSO being used to choose appropriate dataset parameters for increased forecasting accuracy.

In this strategy, BPSO is used to choose the best feature subset for training the SVM model. The goal of BPSO is to increase the accuracy of the SVM model. The method starts with a population of particles, each representing a possible feature subset. Particles alter their placements via iterative updates depending on their own fitness, the fitness of their neighbors, and the fitness of the globally best particle. The method ends when it reaches a stopping condition, such as a predetermined maximum number of iterations or a minimum change in particle fitness.

The findings show that

- the BPSO-SVM model is a reliable computational tool for predicting slope stability. Furthermore, the simulated results show that BPSO-SVM has greater classification accuracy than SVM alone. These findings illustrate the potential and significance of BPSO-SVM as a unique technique to slope stability evaluation that merits future development in environmental sustainability.
- The hybrid BPSO-SVM model enhances slope stability predictions, contributing to sustainable land use and disaster risk reduction. Improved accuracy enables optimized infrastructure planning, early landslide warnings, and efficient resource allocation, minimizing environmental disruptions such as deforestation and soil erosion. By integrating AI method supports climate change adaptation, ecological conservation, and data-driven policymaking. Ultimately, the model promotes safer, more sustainable development, reducing landslide-related risks and fostering long-term environmental resilience.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

1. Duncan, J. M. (1996). State of the art: limit equilibrium and finite-element analysis of slopes. Journal of Geotechnical engineering, 122(7), 577-596.https://doi.org/10.1061/(ASCE)0733-9410(1996)122:7(577)
2. Luo, H., Zeng, C., He, X., Li, W., & Li, Q. (2020, October). Analysis of the Finite Element Strength Reduction Method to Slope Stability with a thin weak layer. In IOP Conference Series: Earth and Environmental Science (Vol. 570, No. 6, p. 062014). https://doi.org/10.1088/1755-1315/570/6/062014
3. Sungkar, M., Munirwansyah, M., Munirwan, R. P., & Safrina, D. (2020, September). Slope stability analysis using Bishop and finite element methods. In IOP conference series: materials science and engineering (Vol. 933, No. 1, p. 012035). IOP Publishing.
4. Huang, Y. H. (2014, February). Slope stability analysis by the limit equilibrium method: Fundamentals and methods. American Society of Civil Engineers. https://doi.org/10.1061/9780784412886
5. Mebrahtu, T. K., Heinze, T., Wohnlich, S., & Alber, M. (2022). Slope stability analysis of deep-seated landslides using limit equilibrium and finite element methods in Debre Sina area, Ethiopia. Bulletin of Engineering Geology and the Environment, 81(10), 403.https://doi.org/10.1007/s10064-022-02906-6
6. Zhou, Y., Zou, P., Wang, F., Liu, Z., Hu, W., & Ma, Z. (2020). Study on high and steep slope stability and slope angle optimization of open-pit based on limit equilibrium and numerical simulation. Geotechnical and Geological Engineering, 38(6), 5737-5753.https://doi.org/10.1007/s10706-020-01390-7
7. Chen, X., Ren, J., & Liu, J. (2017). Analysis of slope stability and software development based on single-grid and two-grid finite element methods. Geotechnical and Geological Engineering, 35, 1369-1382.https://doi.org/10.1007/s10706-017-0182-3

8. Vieira, C. S., Lopes, M. D. L., & Caldeira, L. M. (2013). Limit equilibrium analyses for internal design of geosynthetic reinforced slopes: influence of potential failure surface and strength distribution. Geotechnical and Geological Engineering, 31, 1123-1135.https://doi.org/10.1007/s10706-013-9639-1
9. Rathod, G. W., & Rao, K. S. (2012). Finite element and reliability analyses for slope stability of Subansiri Lower Hydroelectric Project: a case study. Geotechnical and Geological Engineering, 30, 233-252.https://doi.org/10.1007/s10706-011-9465-2
10. Bhandary, R. P., Krishnamoorthy, A., & Rao, A. U. (2019). Stability analysis of slopes using finite element method and genetic algorithm. Geotechnical and Geological Engineering, 37, 1877-1889.https://doi.org/10.1007/s10706-018-0730-5
11. Sun, J., Li, J., & Liu, Q. (2008). Search for critical slip surface in slope stability analysis by spline-based GA method. Journal of geotechnical and geoenvironmental engineering, 134(2), 252-256.https://doi.org/10.1061/(ASCE)1090-0241(2008)134:2(252)
12. Zhang, Q., Wang, Z., & Xia, X. (2012). Interface stress element method and its application in analysis of anti-sliding stability of gravity dam. Science China Technological Sciences, 55, 3285-3291.https://doi.org/10.1007/s11431-012-5059-3
13. Liang, S., Ren, X., & Li, J. (2013). A random medium model for simulation of concrete failure. Science China Technological Sciences, 56, 1273-1281.https://doi.org/10.1007/s11431-013-5200-y
14. Rong, G., Huang, K., Zhou, C., Wang, X., & Peng, J. (2012). A new constitutive law for the nonlinear normal deformation of rock joints under normal load. Science China Technological Sciences, 55, 555-567.https://doi.org/10.1007/s11431-011-4654-z
15. Hammouri, N. A., Malkawi, A. I. H., & Yamin, M. M. (2008). Stability analysis of slopes using the finite element method and limiting equilibrium approach. Bulletin of Engineering Geology and the Environment, 67, 471-478.https://doi.org/10.1007/s10064-008-0156-z
16. Arvin, M. R., Zakeri, A., & Bahmani Shoorijeh, M. (2019). Using finite element strength reduction method for stability analysis of geocell-reinforced slopes. Geotechnical and Geological Engineering, 37, 1453-1467.https://doi.org/10.1007/s10706-018-0699-0
17. Liu, F. (2020). Stability analysis of geotechnical slope based on strength reduction method. Geotechnical and Geological Engineering, 38(4), 3653-3665.https://doi.org/10.1007/s10706-020-01243-3
18. Chen, X., Ren, J., Wang, D., Lyu, Y., & Zhang, H. (2019). A generalized strength reduction concept and its applications to geotechnical stability analysis. Geotechnical and Geological Engineering, 37, 2409-2424.https://doi.org/10.1007/s10706-018-00765-1
19. Feizizadeh, B., Roodposhti, M. S., Blaschke, T., & Aryal, J. (2017). Comparing GIS-based support vector machine kernel functions for landslide susceptibility mapping. Arabian Journal of Geosciences, 10, 1-13.https://doi.org/10.1007/s12517-017-2918-z
20. Hong, H., Pradhan, B., Bui, D. T., Xu, C., Youssef, A. M., & Chen, W. (2017). Comparison of four kernel functions used in support vector machines for landslide susceptibility mapping: a case study at Suichuan area (China). Geomatics, Natural Hazards and Risk, 8(2), 544-569.https://doi.org/10.1080/19475705.2016.1250112
21. Zhao, S., & Zhao, Z. (2021). A Comparative Study of Landslide Susceptibility Mapping Using SVM and PSO-SVM Models Based on Grid and Slope Units. Mathematical problems in Engineering, 2021(1), 8854606.https://doi.org/10.1155/2021/8854606
22. Ballabio, C., & Sterlacchini, S. (2012). Support vector machines for landslide susceptibility mapping: the Staffora River Basin case study, Italy. Mathematical geosciences, 44, 47-70.https://doi.org/10.1007/s11004-011-9379-9
23. Shahin, M. A., Maier, H. R., & Jaksa, M. B. (2002). Predicting settlement of shallow foundations using neural networks. Journal of geotechnical and geoenvironmental engineering, 128(9), 785-793.https://doi.org/10.1061/(ASCE)1090-0241(2002)128:9(785)
24. Shahin, M. A., Maier, H. R., & Jaksa, M. B. (2003). Settlement prediction of shallow foundations on granular soils using B-spline neurofuzzy models. Computers and Geotechnics, 30(8), 637-647.https://doi.org/10.1016/j.compgeo.2003.09.004
25. Sharma, S., Venkateswarlu, H., & Hegde, A. (2019). Application of machine learning techniques for predicting the dynamic response of geogrid reinforced foundation beds. Geotechnical and Geological Engineering, 37, 4845-4864.https://doi.org/10.1007/s10706-019-00945-7
26. Kumar, M., & Samui, P. (2019). Reliability analysis of pile foundation using ELM and MARS. Geotechnical and geological engineering, 37, 3447-3457.https://doi.org/10.1007/s10706-018-00777-x
27. Samui, P. (2012). Application of relevance vector machine for prediction of ultimate capacity of driven piles in cohesionless soils. Geotechnical and Geological Engineering, 30, 1261-1270.https://doi.org/10.1007/s10706-012-9539-9
28. Samui, P. (2011). Prediction of pile bearing capacity using support vector machine. International Journal of Geotechnical Engineering, 5(1), 95-102.https://doi.org/10.3328/IJGE.2011.05.01.95-102
29. Lee, I. M., & Lee, J. H. (1996). Prediction of pile bearing capacity using artificial neural networks. Computers and geotechnics, 18(3), 189-200.https://doi.org/10.1016/0266-352X(95)00027-8

30. Besalatpour, A., Hajabbasi, M. A., Ayoubi, S., Gharipour, A., & Jazi, A. Y. (2012). Prediction of soil physical properties by optimized support vector machines. International Agrophysics, 26(2).
31. Suman, S., Mahamaya, M., & Das, S. K. (2016). Prediction of maximum dry density and unconfined compressive strength of cement stabilised soil using artificial intelligence techniques. International Journal of Geosynthetics and Ground Engineering, 2, 1-11.https://doi.org/10.1007/s40891-016-0051-9
32. Rashid, H. M. A., Sufyan, M., Ismail, A., & Waqas, U. (2023). Experimental Characterization-Based Machine Learning Modeling for the Estimation of Geotechnical Properties of Clay Liners. Geotechnical and Geological Engineering, 41(2), 1503-1519.https://doi.org/10.1007/s10706-022-02350-z
33. Liu, Y., Wang, H., Zhang, H., & Liber, K. (2016). A comprehensive support vector machine-based classification model for soil quality assessment. Soil and Tillage Research, 155, 19-26.https://doi.org/10.1016/j.still.2015.07.006
34. Talamkhani, S., Naeini, S. A., & Ardakani, A. (2023). Prediction of static liquefaction susceptibility of sands containing plastic fines using machine learning techniques. Geotechnical and Geological Engineering, 41(5), 3057-3074.https://doi.org/10.1007/s10706-023-02444-2
35. Lee, C. Y., & Chern, S. G. (2013). Application of a support vector machine for liquefaction assessment. Journal of Marine Science and Technology, 21(3), 10.
36. Hanandeh, S. M., Al-Bodour, W. A., & Hajij, M. M. (2022). A comparative study of soil liquefaction assessment using machine learning models. Geotechnical and Geological Engineering, 40(9), 4721-4734.https://doi.org/10.1007/s10706-022-02180-z
37. Kumar, D., Samui, P., Kim, D., & Singh, A. (2021). A novel methodology to classify soil liquefaction using deep learning. Geotechnical and Geological Engineering, 39, 1049-1058.https://doi.org/10.1007/s10706-020-01544-7
38. Qin, W., & Fan, G. (2023). Estimation and predicting of soil water characteristic curve using the support vector machine method. Earth Science Informatics, 16(1), 1061-1072.https://doi.org/10.1007/s12145-022-00909-1
39. Zhang, Y., Qiu, J., Zhang, Y., & Xie, Y. (2021). The adoption of a support vector machine optimized by GWO to the prediction of soil liquefaction. Environmental Earth Sciences, 80, 1-9.https://doi.org/10.1007/s12665-021-09648-w
40. Rahbarzare, A., & Azadi, M. (2019). Improving prediction of soil liquefaction using hybrid optimization algorithms and a fuzzy support vector machine. Bulletin of Engineering Geology and the Environment, 78(7), 4977-4987.
41. Yang, R., & Yuan, S. (2022). Monitoring and prediction of highway foundation settlement based on particle swarm optimization and support vector machine. Mathematical Problems in Engineering, 2022(1), 2754965.https://doi.org/10.1155/2022/2754965
42. Kalatehjari, R., A Rashid, A. S., Ali, N., & Hajihassani, M. (2014). The Contribution of Particle Swarm Optimization to Three-Dimensional Slope Stability Analysis. The Scientific World Journal, 2014(1), 973093.https://doi.org/10.1155/2014/973093
43. Sadoghi Yazdi, J., Kalantary, F., & Sadoghi Yazdi, H. (2012). Calibration of soil model parameters using particle swarm optimization. International Journal of Geomechanics, 12(3), 229-238.https://doi.org/10.1061/(ASCE)GM.1943-5622.0000142
44. Zou, J., Chen, H., Jiang, Y., Zhang, W., & Liu, A. (2023). An effective method for real-time estimation of slope stability with numerical back analysis based on particle swarm optimization. Applied Rheology, 33(1), 20220143.https://doi.org/10.1515/arh-2022-0143
45. Khajehzadeh, M., Taha, M. R., El-Shafie, A., & Eslami, M. (2014). Stability assessment of earth slope using modified particle swarm optimization. Journal of the Chinese Institute of Engineers, 37(1), 79-87.https://doi.org/10.1080/02533839.2012.757041
46. Pham, B. T., Qi, C., Ho, L. S., Nguyen-Thoi, T., Al-Ansari, N., Nguyen, M. D., Nguyen, H. D., Ly, H. B., Le, H. V., & Prakash, I. (2020). A novel hybrid soft computing model using random forest and particle swarm optimization for estimation of undrained shear strength of soil. Sustainability, 12(6), 2218.https://doi.org/10.3390/su12062218
47. Xue, X., Yang, X., & Chen, X. (2014). Application of a support vector machine for prediction of slope stability. Science China Technological Sciences, 57, 2379-2386.https://doi.org/10.1007/s11431-014-5699-6
48. Xue, X. (2017). Prediction of slope stability based on hybrid PSO and LSSVM. Journal of Computing in Civil Engineering, 31(1), 04016041.https://doi.org/10.1061/(ASCE)CP.1943-5487.0000607
49. Liu, Z., Shao, J., Xu, W., & Xu, F. (2014). Comprehensive stability evaluation of rock slope using the cloud model-based approach. Rock Mechanics and Rock Engineering, 47, 2239-2252.https://doi.org/10.1007/s00603-013-0507-3
50. Li, S., Zhao, H. B., & Ru, Z. (2013). Slope reliability analysis by updated support vector machine and Monte Carlo simulation. Natural Hazards, 65, 707-722.https://doi.org/10.1007/s11069-012-0396-x
51. Cortes, C. (1995). Support-Vector Networks. Machine Learning.https://doi.org/10.1007/BF00994018

52. Vapnik, V. N., & Chervonenkis, A. Y. (1964). On a perceptron class. Avtomatika i Telemkhanika, 25(1), 112-120.
53. 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).https://doi.org/10.1145/130385.130401
54. Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers, 10(3), 61-74.
55. Joachims, T. (2006, August). Training linear SVMs in linear time. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 217-226).https://doi.org/10.1145/1150402.1150429
56. Khanesar, M. A., Teshnehlab, M., & Shoorehdeli, M. A. (2007, June). A novel binary particle swarm optimization. In 2007 Mediterranean conference on control & automation (pp. 1-6). IEEE.
57. Manouchehrian, A., Gholamnejad, J., & Sharifzadeh, M. (2014). Development of a model for analysis of slope stability for circular mode failure using genetic algorithm. Environmental Earth Sciences, 71, 1267-1277.https://doi.org/10.1007/s12665-013-25318.