J-Researchers

Journal of Civil Engineering Researchers

Journal homepage: www.journals-researchers.com

Optimizing Slope Stability Assessment Using Hybrid BPSO - SVC with Kernel Function Evaluation

Saurabh Kumar Anuragi,[a,*]

[a] Department of Civil Engineering, Maulana Azad National Institute of Technology, Bhopal, India

ABSTRACT

The complex nature of slope engineering presents considerable challenges in accurately predicting slope stability using traditional methodologies. Due to the serious implications that can arise from slope failures, it is crucial to implement the most effective techniques for assessing slope stability. This study investigates a hybrid approach that integrates BPSO with SVC to enhance predictive accuracy in slope stability assessment. The methodology employs BPSO to optimize the selection of features that are critical to the prediction process. Additionally, grid search technique is utilized for fine-tuning the hyperparameters of the SVC. The research evaluates the performance of three SVC kernel functions: linear, polynomial and rbf. For the predictive analysis, six features identified as potentially influential were selected: height of the slope (H), pore water ratio (ru), unit weight of the soil (ϒ), cohesion of the soil (c), slope angle (β), and angle of internal friction (ϕ). To enhance the generalization capability of the classification models, a 5-fold cross-validation (CV) approach was implemented. The effectiveness of the models was evaluated using various metrics, including the area under the curve (AUC) and overall accuracy of the predictions. The findings of the study indicate that the hybrid approach, particularly the SVC employing the rbf kernel, significantly outperformed the other models in terms of prediction accuracy, achieving an AUC of 0.735 and an accuracy rate of 0.725. This underscores the potential of the proposed hybrid method as a valuable tool for accurately predicting slope stability and mitigating risks associated with slope failures in engineering applications.

ARTICLE INFO

Received: December 22, 2024
Accepted: January 05, 2025

Keywords:
Support vector classifier
Machine learning
Slope stability
Binary particle swarm
Grid search

DOI: 10.61186/JCER.7.1.1
DOR: 20.1001.1.22516530.1399.11.4.1.1

1. Introduction

Slope stability evaluation represents a critical area of research within the field of slope engineering, as it directly influences the safety and effectiveness of various construction and geological projects. The processes underlying slope deformation and failure are inherently complex and involve intricate geological mechanisms. A multitude of factors contribute to slope stability, many of which are uncertain in nature, making it challenging to conduct accurate evaluations using conventional theoretical analyses and numerical methodologies. Traditional techniques, such as the finite element method [1], discontinuous deformation analysis [2], and limit equilibrium method [3], often fall short in accounting for the intricacies involved in slope behavior. Slope

[*] Corresponding author. Tel.: +91-94069-20449; e-mail: saurabh.1399@gmail.com.

engineering itself is characterized as a complex, non-linear, and dynamic system that is susceptible to various uncertainties. These uncertainties manifest through geological and engineering factors that exhibit randomness, fuzziness, and variability, all of which significantly influence the overall stability of slopes. One of the key challenges in analyzing slope stability is the highly non-linear relationship that exists between slope stability and the myriad influencing factors. Conventional deterministic approaches often fail to capture this non-linearity, leading to potentially inaccurate assessments. Consequently, there is a growing recognition within the research community of the necessity to move beyond traditional deterministic models. This shift aims to embrace a more holistic understanding of the uncertainties associated with the diverse parameters impacting slope stability [4–6]. Given the complex nature of slope engineering, it is vital to incorporate a broad spectrum of geological and engineering considerations, including the unpredictability and variability of these factors, when assessing stability. This evolving perspective emphasizes the urgency for researchers to adopt advanced methodologies that can systematically account for these uncertainties. Ongoing research efforts aim to refine numerical and analytical modeling techniques to better predict possible slope behaviors. By enhancing predictive accuracy, these efforts not only aim to improve understanding of slope stability but also seek to minimize potential losses and inform appropriate preventive measures.

With the advancement of computational techniques, there is a growing opportunity for researchers to implement diverse machine learning methods as alternative approaches for slope stability analysis. By evaluating critical parameters such as slope geometry and material properties, these techniques have the potential to deliver valuable insights and significantly enhance the accuracy of slope stability assessments. Nanehkaran et al. [7] conducted a comparative study evaluating various machine learning techniques for slope stability prediction. Their research included random forest, multilayer perceptron, support vector machines, and decision trees. Similarly, Bui et al. [8] employed five distinct machine learning approaches, namely support vector regression, Gaussian process regression, multilayer perceptron, multiple linear regression, and simple linear regression. Mahmoodzadeh et al. [9] proposed six machine learning techniques for factor of safety (FOS) prediction, which included deep neural networks, Gaussian process regression, support vector regression, k-nearest neighbors, long short-term memory, and decision trees. Nanehkaran et al. [10] further compared five machine learning techniques for FOS prediction, specifically k-nearest neighbors, support vector machines, decision trees, multilayer perceptron, and random forest. Moayedi et al. [11] conducted a comparative analysis among seven machine learning techniques for FOS prediction, including an improved support vector machine utilizing sequential minimal optimization, multiple linear regression, radial basis function regression, random tree, lazy k-nearest neighbor, multilayer perceptron, and random forest. Bai et al. [12] performed a comparative study among eight machine learning techniques for FOS prediction, which comprised gradient boosting decision tree, k-nearest neighbor algorithm, decision tree, guided clustering algorithm, artificial neural network, support vector machine, random forest, and AdaBoost algorithm.

Hybrid models are gaining popularity among researchers due to their ability to enhance accuracy and reliability by combining multiple algorithms. These models leverage the strengths of individual algorithms and compensate for their weaknesses, leading to improved performance across various datasets [13, 14]. For example, integrating optimization techniques like Binary Particle Swarm Optimization (BPSO), Genetic Optimization, Grid Search, Random Search etc. with machine learning classifiers such as Support Vector Classifiers (SVC), Random Forest (RF), XGBoost, CatBoost etc. allows for efficient feature selection while maintaining high prediction accuracy. Luo et al. [15] conducted an extensive slope stability analysis comparing the performance of three standalone machine learning algorithms—k-nearest neighbor (KNN), support vector machine (SVM), and classification and regression tree (CART)—with a hybrid algorithm, the particle swarm optimization-cubist algorithm (PSO-CA). Their study highlighted the potential advantages of integrating optimization techniques with machine learning models to enhance predictive accuracy. Koopialipoor et al. [16] investigated the efficacy of four hybrid machine learning techniques—genetic algorithm-artificial neural network (GA-ANN), artificial bee colony-artificial neural network (ABC-ANN), imperialist competitive algorithm-artificial neural network (ICA-ANN), and particle swarm optimization-artificial neural network (PSO-ANN)—for slope stability prediction. Their research demonstrated that hybrid models could surpass standalone methods by optimizing the training process and improving prediction reliability. Pham et al. [17] performed a comprehensive comparative study of eight machine learning models—decision tree (DT), k-nearest neighbor (KNN), artificial neural network (ANN), Gaussian process (GP), Gaussian naive Bayes (GNB), quadratic discriminant analysis (QDA), support vector machine (SVM), and stochastic gradient descent (SGD)—as well as their combinations in ensemble approaches for slope stability prediction. Their findings underscored the effectiveness of ensemble learning in capturing complex relationships within data. Gordan et al. [18] compared artificial neural networks (ANN) with a particle swarm optimization-enhanced artificial neural network (PSO-ANN) for predicting the factor of safety (FOS) in homogeneous slopes. Their findings emphasized the superior performance of the hybrid PSO-ANN model in improving prediction accuracy. Zhang et al. [19] evaluated the performance of four machine learning models—XGBoost, support vector machine (SVM), random forest (RF), and logistic regression (LR)—in the context of slope stability prediction. Their results illustrated the applicability of contemporary algorithms, such as XGBoost, in achieving high predictive accuracy. Kardani et al. [20] developed a robust hybrid stacking model utilizing eleven machine learning techniques, including random forest (RF), decision tree (DT), k-nearest neighbor (KNN), extreme gradient boosting (XGB), logistic regression (LR), naive Bayes (NB), multilayer perceptron artificial neural networks (MLPANN), bagging classifier (BC), linear discriminant analysis (LDA), support vector classifier (SVC), and extremely randomized trees (ETs). The artificial bee colony (ABC) algorithm was employed to determine the optimal combination of base classifiers, leading to significant enhancements in the model's predictive performance. Qi et al. [21] introduced a hybrid model that integrates the firefly algorithm with six machine learning techniques—gradient boosting machine (GBM), logistic regression (LR), support vector machine (SVM), random forest (RF), multilayer perceptron neural network (MLP), and decision tree (DT)—to evaluate their effectiveness in precise slope stability prediction. Their approach underscored the advantages of metaheuristic algorithms in improving model accuracy and reliability. Zhou et al. [22] applied the gradient boosting machine (GBM) algorithm to slope stability prediction, demonstrating its effectiveness in managing complex datasets and providing accurate predictions. These collective studies reflect a growing trend in the use of advanced and hybrid machine learning models within geotechnical engineering, highlighting their potential to effectively address the challenges associated with slope stability analysis and prediction. These approaches reduce computational complexity and enhance the model's ability to generalize to unseen data. The rise of computational resources and advanced frameworks has made hybrid models increasingly applicable in real-world geotechnical problems, making them essential tools for predictive modeling in complex engineering scenarios.

The aforementioned machine learning models discussed play a vital role in advancing our understanding of slope behavior and the complex interactions that influence slope stability. However, the complexity of slope stability issues can vary, even when utilizing the same dataset. This variability is attributed to the inherent limitations of each model, as different algorithms operate under distinct assumptions and capabilities. The primary objective of slope stability analysis is to achieve accurate and reliable predictions, which necessitates the development and application of more sophisticated machine learning algorithms. To effectively address these challenges, it is essential to identify and utilize advanced hybrid machine learning algorithms that can provide better outcomes compared to traditional standalone models. The findings from the studies reviewed indicate that hybrid learning algorithms, which combine machine learning with optimization techniques, present a promising avenue for predicting slope stability. Nonetheless, there is a noticeable gap in the literature regarding the focused investigation of the various features influencing slope stability, which warrants thorough investigation. A detailed analysis of model behavior using different feature combinations can yield valuable insights into the dynamics of slope stability, ultimately enhancing model robustness. Optimization algorithms are instrumental in this regard, as they can facilitate the selection of the most significant features and refine model parameters to maximize performance. To further advance this field, future research should prioritize the exploration of innovative hybrid learning algorithms that integrate cutting-edge machine learning techniques with optimization strategies. This comprehensive approach can significantly improve predictive capabilities while deepening our understanding of the critical factors affecting slope stability, thereby contributing to the development of safer and more effective geotechnical solutions.

The objective of this study is to evaluate the performance of various kernel functions—specifically linear, polynomial, and radial basis function (RBF)—in the Support Vector Classifier (SVC) integrated with Binary Particle Swarm Optimization (BPSO), with the specific aim of predicting slope stability. The research will thoroughly explore and assess the effectiveness of SVC across different kernel functions and feature combinations. The selection of the SVC model is attributed to its growing prominence and application in engineering disciplines. This study aims to fill the existing gap in literature by providing a comprehensive evaluation of the performance and applicability of these methodologies.

2. Machine Learning Techniques

2.1. Support Vector Classifier

The Support Vector Classifier (SVC) is a sophisticated machine learning technique that utilizes a nonlinear transformation, characterized by an inner product function, to effectively map the input space into a higher-dimensional feature space. The theoretical underpinnings of SVC center on key concepts such as linear separability, decision boundaries, and margin maximization. The seminal work of Vapnik and Chervonenkis (1963) introduced the concept of the Vapnik-Chervonenkis (VC) dimension, which serves as an essential framework for understanding the generalization capabilities of SVC [23]. The VC dimension quantifies the capacity or complexity of a hypothesis space, representing the set of potential decision boundaries that a learning algorithm can derive. Specifically, in the realm of SVC, the VC dimension indicates the maximum number of points that can be perfectly separated by the decision boundary established by the algorithm. This dimension is pivotal in evaluating the balance between model complexity and the capacity to generalize effectively to previously unseen data. Furthermore, the introduction of kernel functions by Boser et al. (1992) [24] significantly enhanced the ability of SVC to process non-linearly separable data. In 1995, Cortes and Vapnik [25] presented the formulation of the Support Vector Classifier, highlighting its ability to identify optimal separating hyperplanes characterized by maximum margins. This formulation incorporates two critical components: the slack variable, ξ, which quantifies the deviation of a data point from the ideal condition, and the penalty factor, C, which delineates the trade-off between the number of misclassifications in the training dataset.

The decision functions for different conditions are:

For Linearly Separable:

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

For Linearly Inseparable;

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

To minimize,

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

For Non-linear Classification:

Linear Kernel:

$$k(x_i x_j) = x_i^T x_j \tag{4}$$

Polynomial Kernel:

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

Radial Basis Function (RBF):

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

Sigmoid Kernel:

$$k(x_i x_j) = \tan(\Upsilon x_i^T x_j + r) \tag{7}$$

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 Y, r and d are kernel parameters.

Numerous studies have concentrated on enhancing the training process and optimizing the performance of support vector classification (SVC). Platt (1999) [26] introduced a sequential minimal optimization algorithm that significantly improved the efficiency of training with large-scale datasets. Furthermore, Joachims (2006) [27] presented the concept of the Budgeted Support Vector Machine, which facilitated quicker training times by selecting a relevant subset of support vectors. Additional research has investigated the utilization of parallel computing, distributed learning, and active learning techniques to accelerate the training process and enhance scalability. From its foundational theoretical principles to its diverse applications across various domains, SVC has demonstrated notable performance and versatility. Despite existing challenges, such as parameter tuning and scalability, ongoing research endeavors aim to address these issues and further enhance the algorithm's effectiveness. An in-depth understanding of the advancements and future directions in SVC will enable researchers to contribute meaningfully to its continued development and explore its potential for tackling complex classification challenges.

2.2. Binary Particle Swarm Optimization (BPSO)

Binary Particle Swarm Optimization (BPSO) is a type of optimization algorithm used to solve binary optimization problems. In binary optimization problems, the goal is to find the binary string that maximizes or minimizes a given objective function. BPSO is a heuristic method that simulates the behavior of a swarm of particles in a multi-dimensional search space. In BPSO, the particles represent positions in a binary space where each element of a particle's position vector can only take on the values of 0 or 1. In other words, $x_i \in B^{n_x}$ or x_{ij} can only be 0 or 1. When a particle's position is updated, it means flipping one or more bits in the binary string representation of the particle. This effectively causes the particle to move to different corners of a hypercube in the binary search space. The flipping of bits can result in the particle moving closer or farther away from the optimal solution.

The binary PSO algorithm begins by randomly generating a population of particles, after which their positions and velocities are initialized. Following this, the fitness function is calculated for each particle, and the best positions of both the individual particles and the entire swarm are updated accordingly. The swarm best position refers to the position of the particle that has the best fitness function value across the entire population. Followed by the velocity and position of each particle is updated using the following equation:

$$\begin{aligned} V_{[i,j]} = w * V_{[i,j]} + c1 * rand\,1 * (p_{best[i,j]} - x_{[i,j]}) \\ + c2 * rand2 * (g_{best[i,j]} - x_{[i,j]}) \\ x_{[i,j]} = 1, if\ rand(0) < sigmoid\ (V_{[i,j]}) \\ x_{[i,j]} = 0, otherwise \end{aligned} \tag{8}$$

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 operates by defining velocities and particle trajectories in terms of the probability of each bit being set to 1 or 0. For instance, a velocity Vij (t) of 0.3 indicates a 30% chance of the corresponding bit being set to 1, and a 70% chance of it being set to 0. To ensure that velocities are interpreted as probabilities, they are typically restricted to the range of [0, 1]. There are various methods for normalizing velocities so that Vij ∈ [0, 1]. One common approach is to divide each v_{ij} by the maximum velocity, $V_{max,j}$. However, if $V_{max,j}$ is large and the actual velocity V_{ij} (t) << $V_{max,j}$ for all time steps, this will reduce the range of velocities, thereby decreasing the chances of a position to change to bit 1. For example, if $V_{max,j} = 10$ and Vij (t) = 5, the normalized velocity V'_{ij} (t) would be 0.5, indicating a mere 50% chance that x_{ij} (t + 1) = 1. Using this normalization method can result in premature convergence to suboptimal solutions, as it limits the exploration abilities of the algorithm.

The velocity normalization is obtained by using sigmoid function

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

Using Eq. (9), the position update changes to

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

Where r_{ij} is a uniform random number in the range [0,1]. The more detailed expressions can be referred at [28].

3. Materials and Methodology

3.1. Data Preprocessing

Developing a classification model for slope stability necessitates the careful selection of features that play a critical role in influencing the outcome. The process of feature selection is pivotal in reducing computational complexity and addressing challenges posed by high-dimensional data. By systematically identifying and retaining the most relevant features, we streamline the learning process, ensuring the model focuses on the essential factors impacting slope stability. This approach not only enhances the efficiency of model training but also mitigates issues like overfitting and redundant computations associated with excessive dimensionality. Ultimately, this strategy contributes to the creation of a robust and reliable classification model capable of accurately predicting slope stability.

Key features such as pore water pressure ratio (ru), slope height (H), unit weight (Y), cohesion (c), slope angle (β), and the angle of internal friction (ϕ) have been widely acknowledged as critical parameters for slope stability prediction. These features capture the essential geotechnical and physical properties influencing slope behavior. For this study, a dataset comprising 444 slope stability cases, categorized into stable (1) and unstable (0) classifications, serves as the foundation for analysis. An illustrative breakdown of these classifications is provided in Figure 1. To further refine the dataset and enhance the analysis, normalization is applied to all features, as described in Equation (11). By scaling the data to a uniform range, this step minimizes potential biases, improves the algorithm's convergence during training, and enhances the model's ability to generalize effectively to unseen cases. Consequently, the normalized dataset facilitates the development of a highly accurate classification model capable of delivering reliable slope stability predictions across diverse scenarios.

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

where, y is a normalized input parameter, x is the original input parameter, x_{max} is the maximum parameter and x_{min} is the minimum parameter.

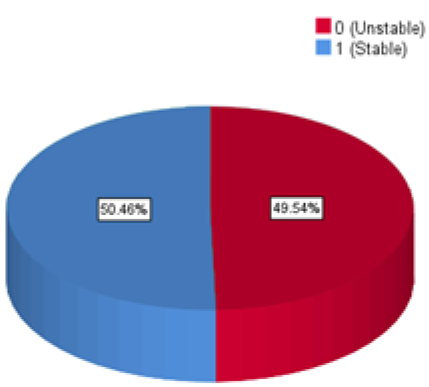

Figure 1. Dataset Pie Chart

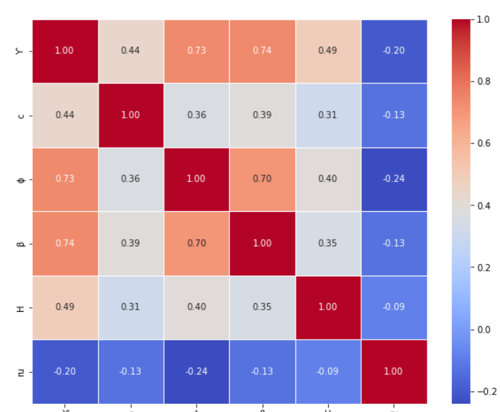

Figure 2. Correlation Matrix of Dataset

The variability and distribution of the input features depicted in Figure 2 demonstrate a strong positive correlation between the features Ƴ and β. Additionally, a significant relationship is noted between ϕ and Ƴ, suggesting that these features are interdependent and may jointly influence predictions related to slope stability. In contrast, the correlation matrix reveals that certain features, such as the ru and c, exhibit weaker correlations with other variables. This suggests that, while these features may have independent effects on slope stability, their interactions with other factors are limited.

Insights derived from the correlation matrix are essential for informed feature selection. Highly correlated features can introduce redundancy within the model, which may elevate computational complexity without substantially enhancing predictive accuracy. Conversely, features with low correlations, though less significant on their own, can still contribute to diversity and enhance the robustness of the model when utilized in combination. Striking a balance in feature selection is imperative for optimizing the feature set to maximize predictive performance while ensuring computational efficiency.

The violin plot illustrated in Figure 3 provides a comprehensive visualization of the distribution and density of the dataset across the various features under consideration. Each violin plot corresponds to a specific feature, with its shape offering valuable insights into the underlying data distribution. The width of the violin at any given point is indicative of the density of data at that value, wider sections signify areas with a higher concentration of data points, while narrower sections reflect regions of lower density. The horizontal line within each violin denotes the median value of the respective feature, serving as a reference for central tendency and aiding in the identification of symmetrical distribution or skewness in the data. Furthermore, the presence of tails at both ends of the violin provides a visual indication of the data's range, thus highlighting any potential outliers or extremes.

The variables Ƴ, ϕ, β, and ru demonstrate a broad distribution, as evidenced by the diverse shapes of their corresponding violins. This broadness suggests a significant dispersion of data points across a wide range of values, indicating high variability in these features, which could be critical in influencing slope stability. In contrast, the variables c and H reveal a more narrowly shaped violin, implying that their data points are densely clustered around specific values. This concentration indicates lower variability and more consistent values compared to the previously mentioned features. The narrow shape highlights a higher frequency of data points near the median, with a reduced occurrence of extreme values.

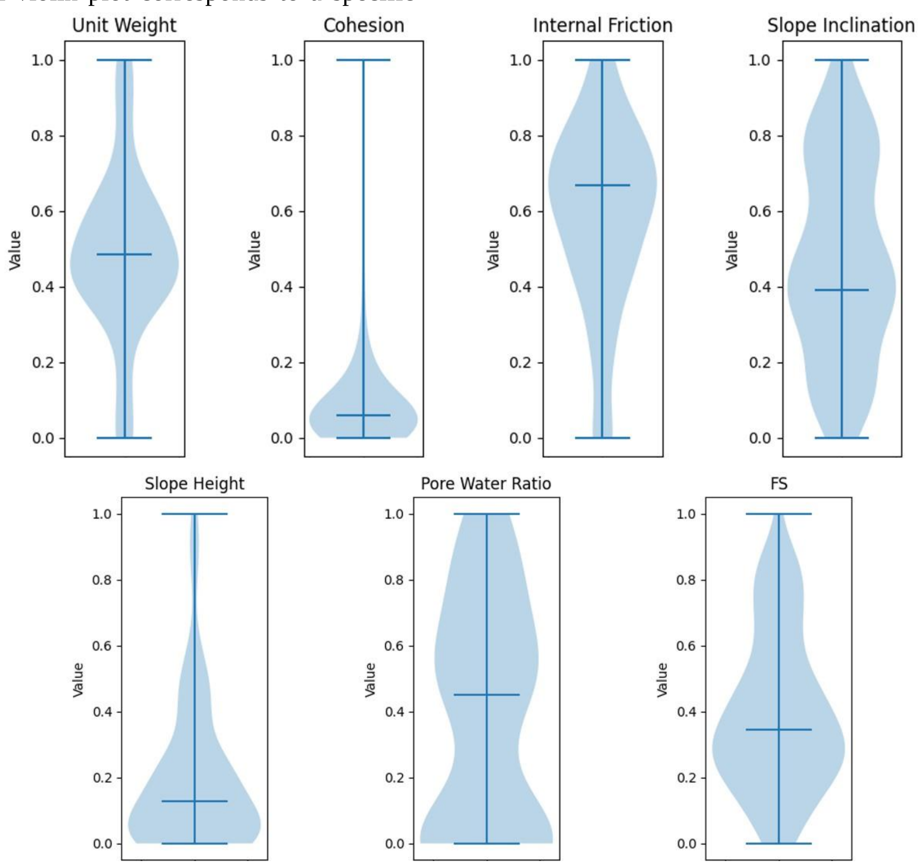

Figure 3. Violin Plots showing distribution of slope cases

This differentiation underscores that features like Y and β may exert a broader influence due to their variability, while features such as c and H may contribute more stable and consistent inputs to the model. These insights are vital for understanding the significance of each feature in the analysis of slope stability and for optimizing overall model performance.

3.2. Model Development and Optimization

This study examines the applicability of SVC utilizing three distinct kernel functions: polynomial, linear, and radial basis function (RBF). The research employs Binary Particle Swarm Optimization (BPSO) for feature selection, as illustrated in Figure 4, which outlines the sequential process of BPSO-SVC. The primary objective of this approach is to enhance the effectiveness of SVC by integrating two critical tasks: the selection of relevant feature subsets from a dataset and the optimization of SVM parameters. In the BPSO framework, each particle in the swarm represents a potential feature subset, with a binary positional vector indicating the presence (1) or absence (0) of specific features. The classification accuracy of each subset is evaluated through a defined fitness function. Importantly, the fitness function that yields the highest accuracy is utilized to assess solutions and update particle positions throughout the iterative process. This methodology aims to identify feature subsets that significantly contribute to achieving precise classification outcomes. Once the best-performing subset is identified, SVC is employed. The optimized features derived from BPSO are subsequently applied to train and test the dataset (Table 1), utilizing the tuned SVC to facilitate improved classification results. The combination of hyperparameters utilized by BPSO are shown in Table 2.

For supervised classification problems, evaluating the performance of classification models on new data is critical to understanding their capacity for generalization.

Tale 1.

Feature optimization using BPSO

Models	Features					
	Y	c	φ	β	H	ru
SVC_rbf	0	1	1	1	0	1
SVC_linear	1	1	1	0	1	0
SVC_poly	0	0	1	0	0	1
0 = Unselected feature			1 = Selected feature			

Table 2

Utilized Hyperparameters for feature optimization using BPSO

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	[1.0]

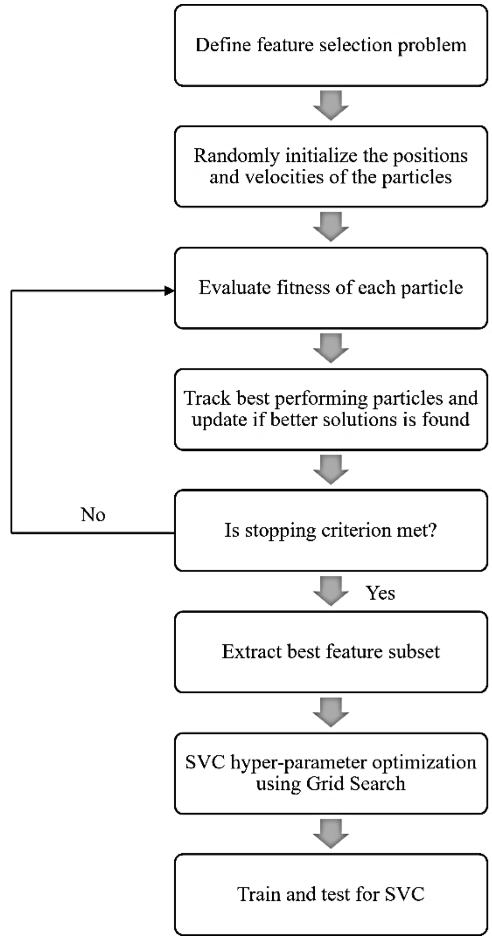

Figure 4. Flow Chart of the BPSO-SVC model

For supervised classification problems, evaluating the performance of classification models on new data is critical to understanding their capacity for generalization. Typically, the dataset is divided into two distinct subsets: the training set and the testing set. The training set, which comprises the majority of the data, is utilized to train the model and optimize hyperparameters. Conversely, the testing set, which constitutes a smaller portion of the dataset, is reserved exclusively for assessing the model's ability to generalize to new, unseen instances. In this study, 70% of the original dataset, amounting to 311 cases, has been designated as the training set, while around 30%, corresponding to 133 cases, has been allocated as the testing set. This division ensures that the model is trained on a diverse range of data while maintaining a separate, independent subset for a rigorous evaluation of its performance on unseen data. The training process for each kernel function involves the exploration of various combinations of hyperparameters, as detailed in the accompanying Table 3. By employing grid search technique, optimal hyperparameters of are identified, facilitating the achievement of the best model performance. These optimal parameters are subsequently applied for making predictions on unseen data, thereby validating the model's effectiveness in real-world applications.

Table 3

SVC optimal hyperparameters using grid search

Model	Hyperparameters	Optimal Hyperparameters
SVC_poly	C = [1 – 200]; step size =5	186
	Degree = [1 – 6]	2
SVC_linear	C = [1 – 200]; step size =5	11
SVC_rbf	C = [1 – 200]; step size =5	41

The SVC model with different kernels is assessed using a technique known as 5-fold cross-validation. This approach enhances the robustness of the model and its ability to generalize to new data. The Area Under the Curve (AUC) metric serves as a comprehensive measure of the model's overall performance across both the training and testing sets. By considering the model's predictive capacity, particularly its ability to discriminate between classes under varying thresholds, the AUC metric enables a thorough examination and validation of the machine learning algorithm's capability to capture underlying patterns and generalize effectively to previously encountered instances.

4. Results and Discussion

The study employs a hybrid approach incorporating BPSO-SVC to assess its efficacy in predicting slope stability. The BPSO algorithm is utilized to determine optimal feature combinations for each SVC kernel function, and subsequent grid search optimization is conducted to fine-tune the hyperparameters of the SVC models. The performance of the classifiers is evaluated using various metrics, including the AUC scores. The findings indicate that the AUC scores for the models are as follows: SVC_linear at 0.671, SVC_poly at 0.681, and SVC_rbf at 0.735 (Figure 5). The observed differences in AUC values can be attributed to variations in the underlying algorithms, model complexity, and each classifier's capacity to capture the relationships between features and the target variable. Notably, SVC_rbf achieved the highest AUC score, signifying its superior discriminatory ability and overall performance in comparison to the other models.

A detailed analysis of the confusion matrices (as illustrated in Figure 6) highlights the misclassification counts for each model: SVC_linear exhibited 45 misclassifications, SVC_poly had 43, while SVC_rbf recorded the lowest number at 35. This reduced error rate

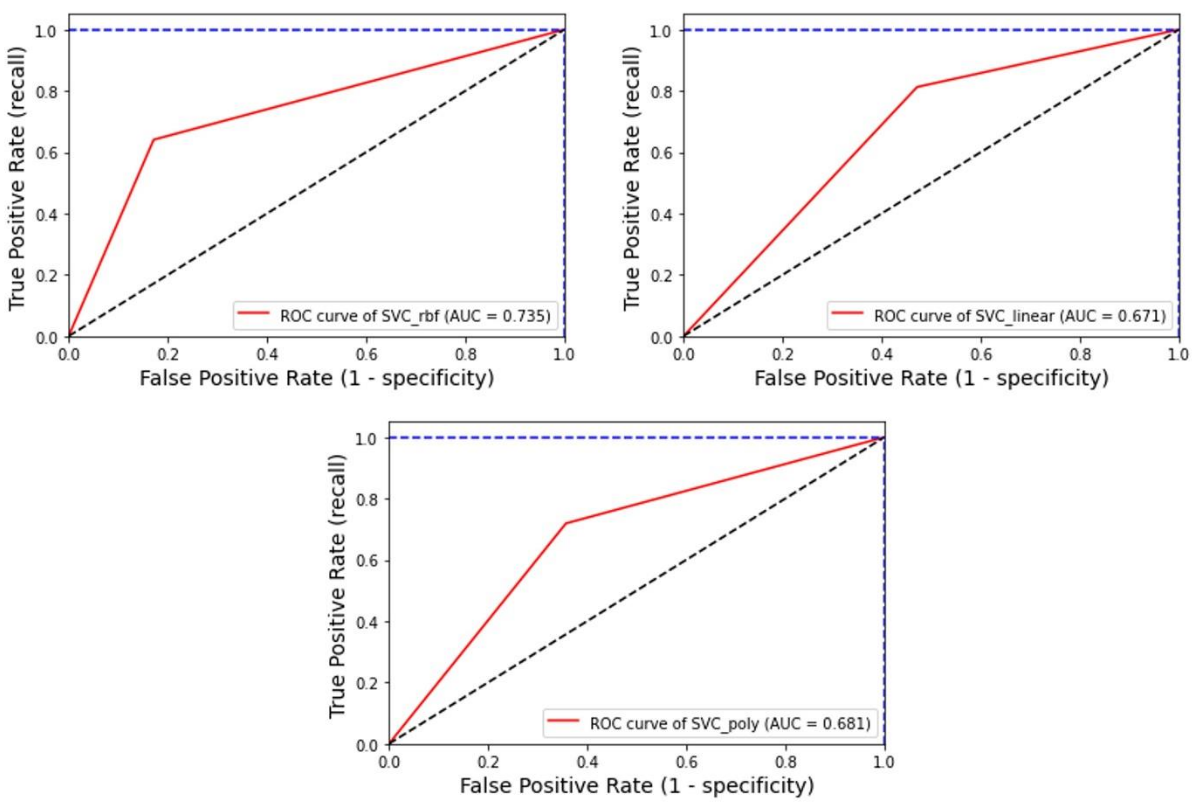

Figure 5. ROC curves of classification models

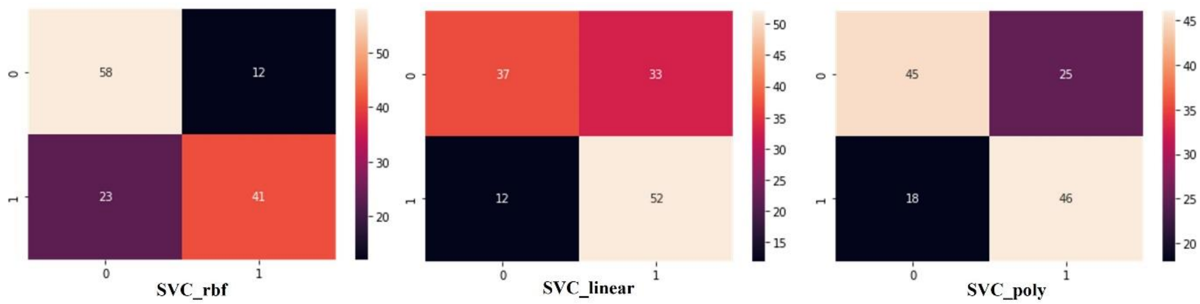

Figure 6. Confusion matrix of classification models

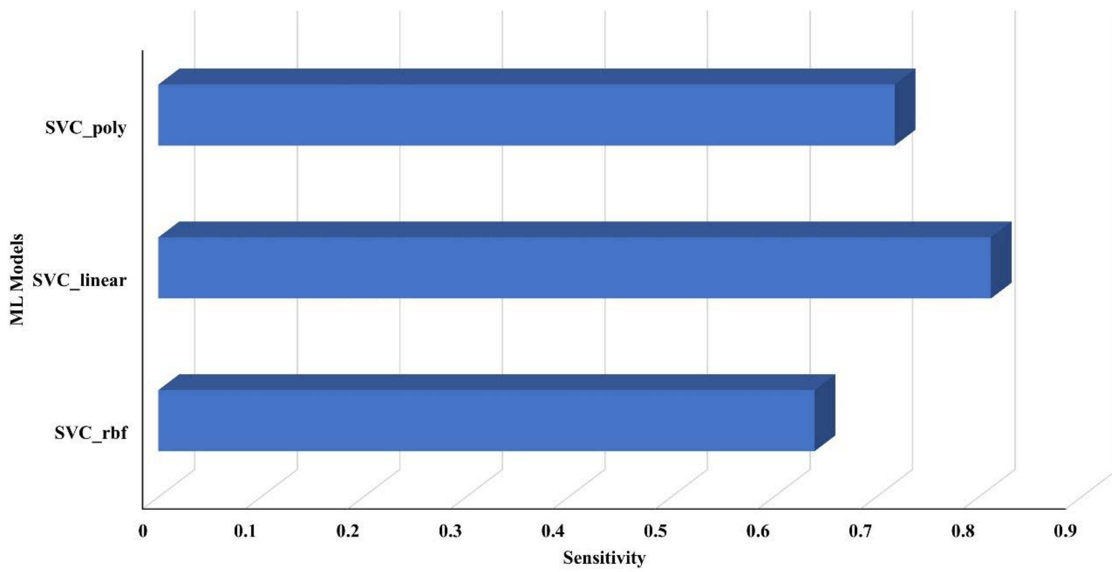

Figure 7. Sensitivity of classification models

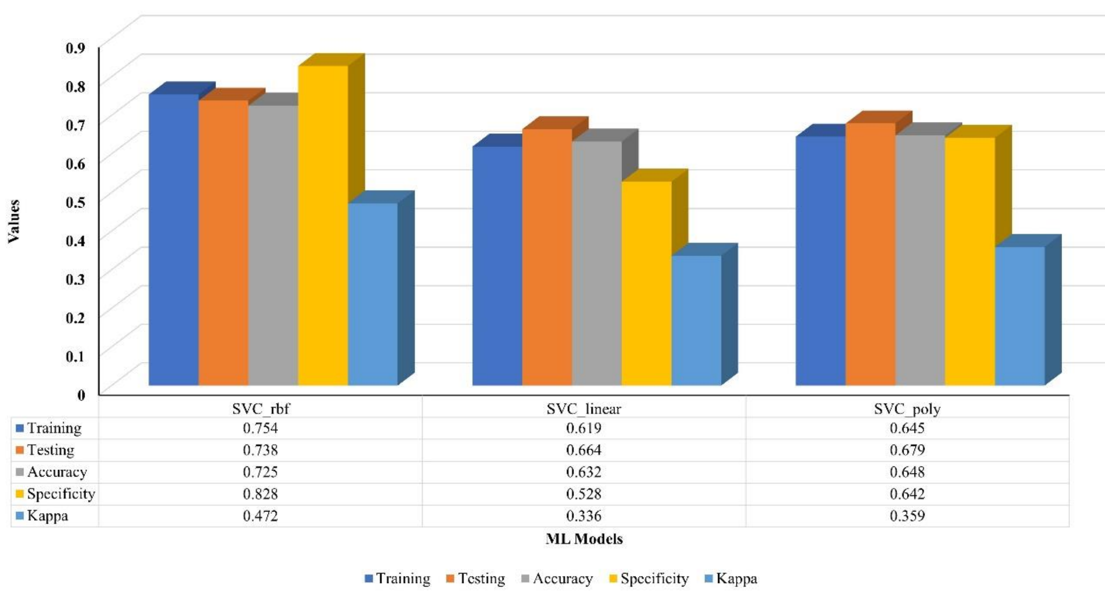

Figure 8. Evaluation metrics of classification models

for SVC_rbf further corroborates its effectiveness in accurately classifying slope stability cases. Furthermore, the sensitivity analysis (depicted in Figure 7) reveals notable performance disparities, with SVC_linear achieving the highest sensitivity score of 0.812, followed by SVC_poly at 0.718, and SVC_rbf at 0.640. Although SVC_linear excels in sensitivity, SVC_rbf demonstrates a more balanced performance across other critical metrics, including specificity (0.828), accuracy (0.725), and kappa (0.472) as shown in Figure 8. Consequently, SVC_rbf emerges as the most robust and reliable model for slope stability classification.

5. Conclusion

This study illustrates the efficacy of a hybrid approach that integrates Binary Particle Swarm Optimization (BPSO) with a Support Vector Classifier (SVC) for predicting slope stability. By employing BPSO to optimize feature selection and utilizing grid search for fine-tuning the hyperparameters of the SVC, the research assessed the performance of three distinct SVC kernel functions: linear, polynomial, and radial basis function (RBF). Notably, the SVC employing the RBF kernel consistently surpassed the other models across key performance metrics, achieving the highest Area Under the Curve (AUC) score of 0.735, the fewest misclassification errors (35), and the highest specificity rate of 0.828. These findings underscore its superior capability to effectively differentiate between stable and unstable slope conditions.

While the SVC with the linear kernel demonstrated the highest sensitivity at 0.812, it exhibited limitations in other metrics such as specificity and overall accuracy, rendering it less suitable for balanced classification tasks. The SVC with the polynomial kernel displayed intermediate performance, with an AUC of 0.681 and a comparatively lower number of misclassifications (43). However, the comprehensive performance profile of the SVC with the RBF kernel—characterized by strong discriminatory power and lower error rates—solidifies its status as the most reliable and robust classifier for this application. This research highlights the potential of hybrid methodologies like BPSO-SVC to advance predictive modeling in slope stability analysis through enhanced optimization of both feature selection and model parameters. Future investigations could benefit from exploring the integration of additional optimization algorithms, incorporating ensemble learning techniques, or evaluating alternative kernel functions to further improve model performance and generalizability. This study lays the groundwork for the application of hybrid approaches in addressing complex geotechnical challenges with greater accuracy and reliability.

References

[1] Duncan, James Michael. "State of the art: limit equilibrium and finite-element analysis of slopes." Journal of Geotechnical engineering 122.7 (1996): 577-596. https://doi.org/10.1061/(ASCE)0733-9410(1996)122:7(577)

[2] Zheng, Fei, et al. "Modified predictor-corrector solution approach for efficient discontinuous deformation analysis of jointed rock masses." International Journal for Numerical and Analytical Methods in Geomechanics 43.2 (2019): 599-624. https://doi.org/10.1002/nag.2881

[3] Liu, S. Y., L. T. Shao, and H. J. Li. "Slope stability analysis using the limit equilibrium method and two finite element methods." Computers and Geotechnics 63 (2015): 291-298. https://doi.org/10.1016/j.compgeo.2014.10.008

[4] Griffiths, D. V., and P. A. Lane. "Slope stability analysis by finite elements." Geotechnique 49.3 (1999): 387-403. https://doi.org/10.1680/geot.1999.49.3.387

[5] Matsui, Tamotsu, and Ka-Ching San. "Finite element slope stability analysis by shear strength reduction technique." Soils and foundations 32.1 (1992): 59-70. https://doi.org/10.3208/sandf1972.32.59

[6] Li, X. "Finite element analysis of slope stability using a nonlinear failure criterion." Computers and Geotechnics 34.3 (2007): 127-136. https://doi.org/10.1016/j.compgeo.2006.11.005

[7] Nanehkaran, Yaser A., et al. "Comparative analysis for slope stability by using machine learning methods." Applied Sciences 13.3 (2023): 1555. https://doi.org/10.3390/app13031555

[8] Tien Bui, Dieu, et al. "Predicting slope stability failure through machine learning paradigms." ISPRS International Journal of Geo-Information 8.9 (2019): 395. https://doi.org/10.3390/ijgi8090395

[9] Mahmoodzadeh, Arsalan, et al. "Prediction of safety factors for slope stability: comparison of machine learning techniques." Natural Hazards (2022): 1-29. https://doi.org/10.1007/s11069-021-05115-8

[10] Ahangari Nanehkaran, Yaser, et al. "Application of machine learning techniques for the estimation of the safety factor in slope stability analysis." Water 14.22 (2022): 3743. https://doi.org/10.3390/w14223743

[11] Moayedi, Hossein, et al. "Machine-learning-based classification approaches toward recognizing slope stability failure." Applied Sciences 9.21 (2019): 4638. https://doi.org/10.3390/app9214638

[12] Bai, Gexue, et al. "Performance evaluation and engineering verification of machine learning based prediction models for slope stability." Applied Sciences 12.15 (2022): 7890. https://doi.org/10.3390/app12157890

[13] Saurabh Kumar Anuragi, D.Kishan, S.K.Saritha. "Comparative Analysis of Ensemble Learning Approaches for Slope Stability Prediction". SSRG International Journal of Civil Engineering 11.5 (2024) :168-180. https://doi.org/10.14445/23488352/IJCE-V11I5P116

[14] Saurabh Kumar Anuragi. "Utilizing Boosting Machine Learning Techniques for Slope Stability Prediction". International Journal of Civil and Structural Engineering Research 12.2 (2024) : 53–58. https://doi.org/10.5281/zenodo.14274474

[15] Luo, Zhenyan, et al. "A novel artificial intelligence technique for analyzing slope stability using PSO-CA model." Engineering with Computers 37 (2021): 533-544. https://doi.org/10.1007/s00366-019-00839-5

[16] Koopialipoor, Mohammadreza, et al. "Applying various hybrid intelligent systems to evaluate and predict slope stability under static and dynamic conditions." Soft Computing 23 (2019): 5913-5929. https://doi.org/10.1007/s00500-018-3253-3

[17] Pham, Khanh, et al. "Ensemble learning-based classification models for slope stability analysis." Catena 196 (2021): 104886. https://doi.org/10.1016/j.catena.2020.104886
[18] Gordan, Behrouz, et al. "Prediction of seismic slope stability through combination of particle swarm optimization and neural network." Engineering with computers 32 (2016): 85-97. https://doi.org/10.1007/s00366-015-0400-7
[19] Zhang, Wengang, et al. "Slope stability prediction using ensemble learning techniques: A case study in Yunyang County, Chongqing, China." Journal of Rock Mechanics and Geotechnical Engineering 14.4 (2022): 1089-1099. https://doi.org/10.1016/j.jrmge.2021.12.011
[20] Kardani, Navid, et al. "Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data." Journal of Rock Mechanics and Geotechnical Engineering 13.1 (2021): 188-201. https://doi.org/10.1016/j.jrmge.2020.05.011
[21] Qi, Chongchong, and Xiaolin Tang. "Slope stability prediction using integrated metaheuristic and machine learning approaches: A comparative study." Computers & Industrial Engineering 118 (2018): 112-122. https://doi.org/10.1016/j.cie.2018.02.028
[22] Zhou, Jian, et al. "Slope stability prediction for circular mode failure using gradient boosting machine approach based on an updated database of case histories." Safety Science 118 (2019): 505-518. https://doi.org/10.1016/j.ssci.2019.05.046
[23] Vapnik, Vladimir N., and A. Ya Chervonenkis. "On a perceptron class." Avtomatika i Telemkhanika 25.1 (1964): 112-120.
[24] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." Proceedings of the fifth annual workshop on Computational learning theory. 1992. https://doi.org/10.1145/130385.130401
[25] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20 (1995): 273-297. https://doi.org/10.1007/BF00994018
[26] Platt, John. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods." Advances in large margin classifiers 10.3 (1999): 61-74.
[27] Joachims, Thorsten. "Training linear SVMs in linear time." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006. https://doi.org/10.1145/1150402.1150429
[28] Khanesar, Mojtaba Ahmadieh, Mohammad Teshnehlab, and Mahdi Aliyari Shoorehdeli. "A novel binary particle swarm optimization." 2007 Mediterranean conference on control & automation. IEEE, 2007. https://doi.org/10.1109/MED.2007.4433821