ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

Research on the Application of Moss Growth Optimization Algorithm in Machine Learning Hyperparameter Tuning: Taking Stroke Risk Assessment as an Example

Wei Guo[1,a], Hao Liu[1,b,*]

[1]*University of Science and Technology Liaoning, Anshan, China*
[a]*weiguo@stu.ustl.edu.cn*, [b]*haoliu@ustl.edu.cn*
**Corresponding author*

Abstract: *Addressing the challenges of hyperparameter optimization in stroke risk models, this study proposes a tuning framework using the Moss Growth Optimization (MGO) algorithm. MGO mimics moss growth mechanisms—spore diffusion, dual reproduction, and cryptobiotic survival—to implement a three-phase strategy combining global exploration, local refinement, and adaptive escape. This enables efficient navigation of high-dimensional, non-convex hyperparameter spaces. Experiments on the Kaggle stroke dataset evaluated eight machine learning models. MGO-optimized XGBoost, AdaBoost, and GBDT achieved near-perfect classification: accuracy ≥99.98% and ROC-AUC=1.0000, outperforming traditional tuning methods. Linear discriminant analysis and KNN showed moderate performance, while simpler architectures (e.g., decision trees) suffered from overfitting. The framework enhances adaptability to complex medical data through balanced global-local search dynamics and improves minority-class recognition in imbalanced datasets. This work establishes a systematic approach for medical ML optimization, advancing technical foundations for precise clinical stroke risk assessment.*

Keywords: *Moss Growth Optimization, Hyperparameter Tuning, Stroke Risk, Machine Learning*

1. Introduction

Stroke, a cerebrovascular disease with high incidence, disability, and mortality rates, imposes significant burdens on individuals and society [1]. Accurate risk assessment enables early intervention to mitigate occurrence and improve prognosis [2], where machine learning demonstrates strong potential through predictive model development using medical data [3].

Hyperparameter configuration critically affects ML model performance, as optimal settings enhance feature extraction and prediction accuracy [4]. Traditional methods like grid/random search face limitations in computational efficiency and local optima trapping, particularly with complex medical datasets [5]. Bio-inspired algorithms offer promising alternatives: genetic algorithms utilize evolutionary operations (selection/crossover/mutation) [6], while particle swarm optimization simulates bird flock foraging through particle coordination [7].The Moss Growth Optimization (MGO) algorithm, a novel bio-inspired method simulating moss propagation strategies, integrates spore dispersal patterns, dual propagation search, and cryptobiotic survival to enable population renewal and environmental adaptation. This grants robust global search capabilities and convergence [8]. While MGO shows efficacy in engineering optimizations (e.g., pressure vessel design [9]), its application to ML hyperparameter tuning—particularly in medical contexts like stroke risk assessment—remains underexplored. This study aims to develop an ML optimization framework for stroke risk evaluation to advance medical AI applications and strengthen stroke prevention strategies.

While existing research confirms MGO's effectiveness in engineering optimizations like pressure vessel design [9], its application to machine learning hyperparameter tuning remains underexplored, particularly in medical contexts like stroke risk assessment. This study seeks to establish an efficient machine learning optimization framework for stroke risk evaluation, aiming to advance medical AI applications and enhance technical support for stroke prevention strategies.

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

2. Related Works

Globally, stroke remains a major threat to human health, necessitating precise risk assessment for early intervention. Machine learning (ML) leverages nonlinear modeling and data mining to advance stroke risk prediction, classification, and clinical decision-making. In risk stratification, Jamthikar [10] integrated traditional risk factors (CRF) and carotid ultrasound phenotypes (CUSIP), optimizing feature combinations via PCA polling to improve accuracy using carotid stenosis as an event-equivalent gold standard. Orfanoudaki [11] proposed a nonlinear Framingham risk score (N-SRS) to uncover risk interactions overlooked by traditional models, though its population generalizability requires further validation. For prognosis prediction, Heo [12] demonstrated that deep neural networks (DNNs) significantly outperformed random forests and logistic regression in predicting 3-month functional outcomes for acute ischemic stroke, supporting personalized treatment. However, multi-center validation and integration of clinical variables are still needed. In data integration, Daidone Mario [13][14] combined heterogeneous data (e.g., electronic health records, imaging, lab indicators) to capture interactions between factors like hypertension and smoking, while incorporating subtle features (e.g., T-wave anomalies, hematocrit). DNNs were employed to detect CT scan abnormalities for emergency triage, though technical and ethical challenges persist. Model optimization studies highlight practical advancements. Li [15] addressed missing data using a random forest-boosting tree hybrid model, enhancing high-risk population identification and cost efficiency. Hassan [16] tackled class imbalance with ensemble strategies (voting/fusion), identifying age, blood glucose, and BMI as key predictors. Dritsas [17] improved minority-class recognition via stacked integration, achieving superior accuracy and recall. Despite progress, gaps remain in hyperparameter optimization and systematic algorithm comparisons. This study applies the Moss Growth Optimization (MGO) algorithm to refine ML hyperparameters, aiming to balance model efficiency and predictive performance for enhanced clinical utility.

3. The principle of Moss Growth Optimization Algorithm

3.1. Algorithm Background

Mosses, an ancient plant group, survive diverse environments through specialized adaptations despite lacking complex structures like flowers, roots, or vascular systems. Their life cycle alternates between photosynthetic gametophytes and dependent sporophytes. Mature sporophytes release spores in stable morning winds, enabling long-distance dispersal, while turbulence causes random dispersion with reduced range. Sexual reproduction requires water for sperm migration to archegonia, producing sporophytes reliant on gametophytes. Environmental resources regulate sporophyte formation frequency. Vegetative reproduction through gametophyte fragments maintains local populations and rapid colonization. This dual strategy balances spore-based expansion with fragment-driven consolidation. Under extreme stress, mosses enter cryptobiosis - a metabolically inactive, glassy state reversed by environmental improvement, enabling survival in hostile conditions.

Drawing from moss adaptation mechanisms, this study develops a machine learning hyperparameter optimization method: Spore-Driven Global Exploration: Simulating wind-dispersed spores, this module systematically explores hyperparameters to identify potential optima while addressing random search inefficiencies. Dual Reproduction Local Optimization: Merging genetic diversity (sexual) and clonal stability (vegetative), it applies controlled perturbations in promising parameter regions to balance search diversity and directional refinement. Cryptobiotic Adaptive Selection: During stagnation, this mechanism dynamically adjusts greedy selection, suppressing dominant solutions to escape local optima and improve robustness in non-convex spaces.

By integrating these mechanisms, the MGO algorithm achieves synergistic global exploration, local refinement, and adaptive escape, offering a novel framework for high-dimensional, multimodal hyperparameter optimization. This approach may enhance model training efficiency and generalization capabilities. The biological foundations discussed here align with principles outlined by Zheng, B. et al. [9]

3.2. Establishment of Model

Based on the moss growth model, MGO consists of four core mechanisms: wind direction determination, spore diffusion search, double reproduction search and cryptobiotic mechanism. Among

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

them, the determination of wind direction is the key mechanism that decides the evolutionary direction of the population.

3.2.1. Determination of Wind Direction

The algorithm draws from moss population dynamics, particularly wind-regulated spore dispersal patterns. Its wind direction strategy adjusts search paths by analyzing individual positions relative to the current best solution, creating a geometric link between individual positions and the global optimum to guide exploration while reducing local optima trapping. The framework follows three biological hypotheses: wind direction remains fixed per iteration; the current best solution equals the global optimum per cycle; wind vectors originate from population-dense areas toward high-fitness individuals.

For each dimension m, the population X is divided into two subsets:

$$\begin{aligned} DX_{m,1} &= \{M_n |\ M_{n,m} \geq M_{\mathbf{best},m}, M_n \in X\}, \\ DX_{m,2} &= \{M_n |\ M_{n,m} < M_{\mathbf{best},m}, M_n \in X\} \end{aligned} \tag{1}$$

Select the one with a larger number of individuals in the population as the subset of the dominant body:

$$Dominate\ X_m = \begin{cases} DX_{m1}\ , & |DX_{m1}| \geq |DX_{m2}| \\ DX_{m2}\ , & |DX_{m1}| < |DX_{m2}| \end{cases} \tag{2}$$

Based on the dimension calculation, select no more than specific non-repetitive dimensions, calculate their intersections, and obtain the dominant body population:

$$\begin{aligned} Dominate\ X &= \left\{ M_n |\ M_n \in \bigcap_{m=1}^{dim} Dominate\ X_{Km}, M_n \in X \right\} \\ \mathbf{Satisfied:} &\begin{cases} \bigcap_{m=1}^{dim} Km = \varnothing \\ \dim = Dim / 4 \end{cases} \end{aligned} \tag{3}$$

Calculate the set of distance vectors between individuals in the dominant population and the optimal individual:

$$Distance\ X = \{M_{\mathbf{best}} - M_n |\ M_n \in Dominate\ X\} \tag{4}$$

The wind direction is the average value of the distance vector:

$$Wind = \frac{1}{num} \sum_{n=1}^{num} dM_n, \quad dM_n \in Distance\ X \tag{5}$$

The algorithm models moss organisms as population-based search agents, simulating spore dispersal dynamics through a wind direction determination model anchored to the fittest individual as the reference origin. An adaptive weighting mechanism balances historical optima and population diversity, generating evolutionary vectors to guide biased stochastic movements that harmonize global exploration with local optimization. This bio-inspired framework enables efficient high-dimensional parameter optimization via coordinated migration-perturbation strategies.

3.2.2. Spore Diffusion Search

The spore diffusion module employs a dynamic step size strategy based on moss spore propagation. It uses two probabilistic search modes: 80% probability for large-step global exploration and 20% probability for small-step local refinement, eliminating fixed-step limitations. A wind parameter E decreasing with iterations is introduced: higher initial values enable wide-range exploration, while reduced values guide focused exploitation of high-quality regions, balancing search efficiency and solution space coverage through adaptive step size and wind force adjustments:

$$M_m^{new} = \begin{cases} M_m + Process\ 1 \cdot Wind, & r_1 > d_1 (Stable) \\ M_m + Process\ 2 \cdot Wind, & r_1 \leq d_1 (Turbulence) \end{cases} \tag{6}$$

Among them, $\mathrm{d}_1 = 0.2, \mathrm{r}_1 \sim (0,1)$, M_m^{new} indicates the diffusion of spores from the m-TH moss individual to generate a new type of moss.

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

3.2.3. Double Reproduction Search

During local refinement, MGO activates dual reproduction (when random value C < 0.8) to enhance solution precision. Sexual reproduction blends current and elite hyperparameters via crossover to generate diverse offspring, while vegetative reproduction fine-tunes random parameter dimensions guided by wind direction vectors. This dual strategy balances population diversity with localized search precision, accelerating convergence toward optimal solutions.

Sexual reproduction generates new individuals by fusing the characteristics of the current individual and the optimal individual M_n^{new}:

$$M_n^{new} = (1-bp)\cdot M_n + bp\cdot M_{\mathbf{best}} \tag{7}$$

$$bp = \begin{cases} 1, & \dfrac{1}{1.5-10\cdot r_5} \geq 0.5 \\ 0, & else \end{cases} \tag{8}$$

This mechanism realizes the dynamic integration of global and local information. Through random decision-making, it flexibly adjusts the inheritance weights of the new individual to the current individual M_n and the optimal individual M_{best}, enhancing the diversity and adaptability of the search process.

Asexual reproduction first randomly selects a dimension m, and then updates this dimension based on the wind direction and step size. The calculation formula for the dimension value $M_{n,m}^{new}$ of the new individual is:

$$M_{n,m}^{new} = M_{\mathbf{best},m} + Process\ 3\cdot Wind_m \tag{9}$$

Among them, $M_{best,m}$ is the m-TH particle in M_{best}.

This strategy guides search direction using wind direction and enhances local search flexibility via random step sizes, enabling fine-grained exploration of optimal neighborhoods and improving local development accuracy.

3.2.4. Cryptobiosis Mechanism

The Cryptobiosis mechanism optimizes greedy selection by preserving historical individual information. When reaching maximum records or iteration end, optimal historical individuals replace current solutions to avoid local optima, improve population quality, and balance exploration/development. Each individual M_n maintains a record list for this process:

$$rM_n = [rM_n^0, rM_n^1, \ldots, rM_n^{rec\_\min\text{-}1}] \tag{10}$$

This list is used to store the historical states of individuals during the iterative process, providing a data basis for the subsequent recovery of the optimal solution. Initial record: $rM_n^0 = M_n$. When the maximum number of records is reached or the iteration ends, traverse the historical records and select the individual with the best fitness rM_n^{best} to replace the current individual. This ensures that when the algorithm encounters local optima or search stagnation, it can redirect search using historical high-quality solutions, enhancing population quality and search efficiency.

4. Experimental Results and Analysis

This study applies the Moss Growth Optimization algorithm to tune hyperparameters for eight machine learning models using the Kaggle stroke risk dataset. The framework initializes continuous/discrete parameter ranges with boundary constraints, generating populations through uniform distribution. Fitness evaluation employs five-fold cross-validation mean accuracy, assigning default values to invalid configurations. The search mechanism combines elite-guided wind direction vectors with dynamic step sizes - broad exploration initially and refined tuning later through decaying Gaussian noise and boundary reflection.

Confusion matrices reveal XGBoost and AdaBoost achieve 100% accuracy, demonstrating perfect classification. GBDT shows two FP errors with otherwise preferable performance. As shown in Fig.1. Linear Discriminant Analysis maintains high accuracy despite minor errors. KNN exhibits balanced FP/FN errors reducing overall accuracy. Naive Bayes and Random Forest display moderate misclassification rates, while Decision Tree shows the highest error frequency across all sample types.

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

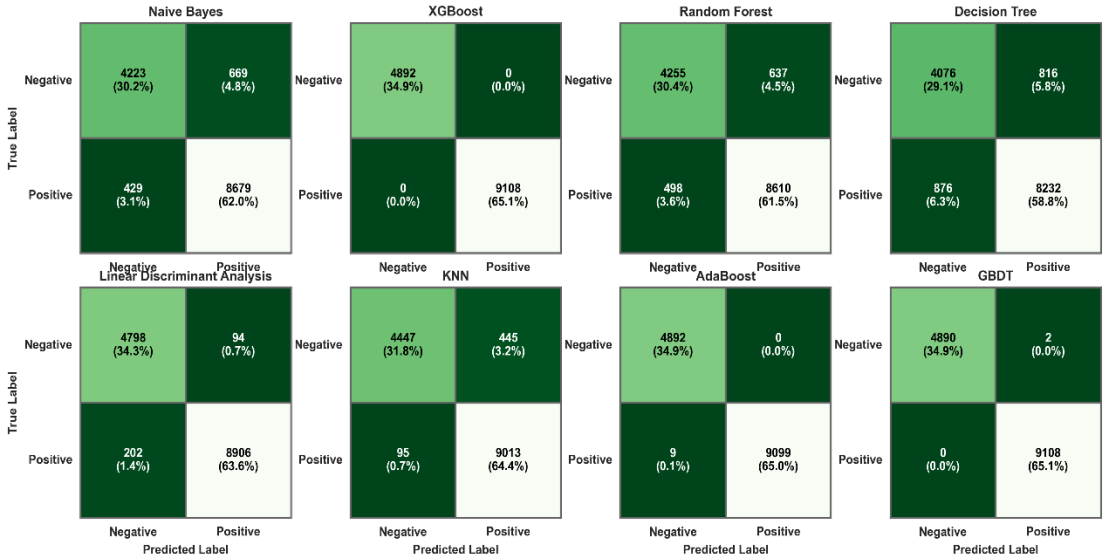

Figure 1: The confusion matrix of the results of each model.

As shown in Fig.2, visual analysis of hyperparameter influences on the LDA model reveals that the solver parameter has minimal effect on cross-validation accuracy, with svd values showing a more concentrated accuracy distribution in the range of 0.9–0.96; the shrinkage parameter is significantly negatively correlated with accuracy, where smaller values help maintain high accuracy but exceed 0.2 leads to a notable decline; additionally, under the tolerance parameter, most search points' accuracy rates are concentrated within 0.96–1.0 without an obvious monotonic change trend, indicating limited influence of this parameter within a certain range.

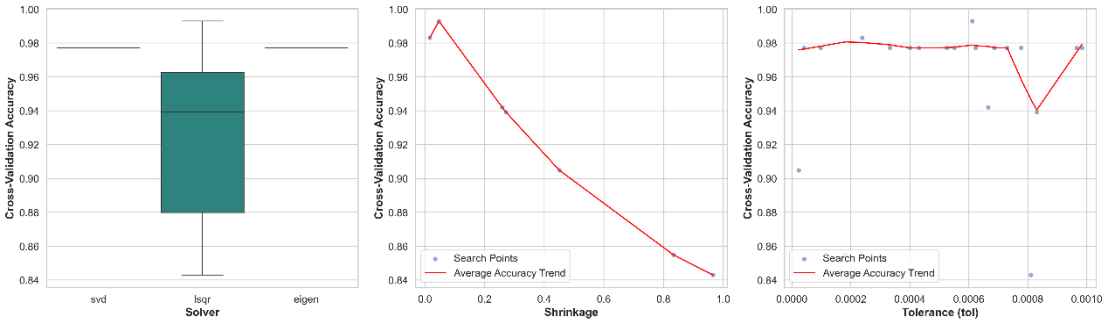

Figure 2: The influence of hyperparameters on the Linear Discriminant Analysis model.

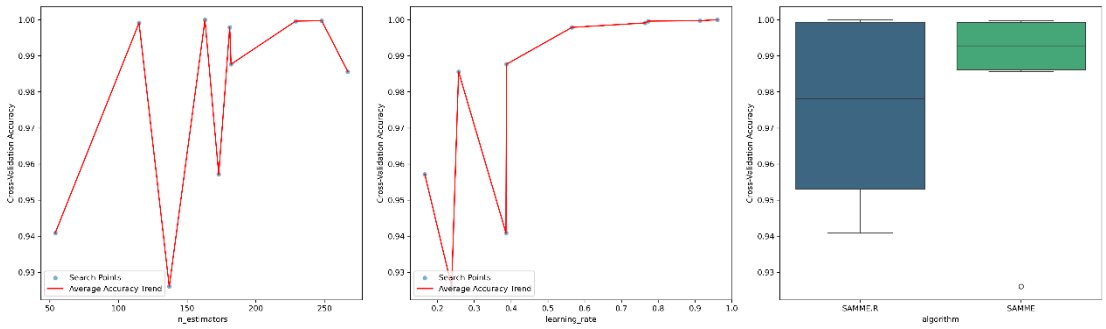

Figure 3: The influence of hyperparameters on the AdaBoost model.

As depicted in Fig.3, visual analysis of hyperparameters in the AdaBoost model shows that as the n_estimators parameter increases, accuracy exhibits fluctuating patterns, reaching peaks at specific values such as 120 and 180 while being relatively lower at 140, indicating an optimal value range; learning_rate shows significant fluctuations between 0.1 and 0.3 but stabilizes at higher values exceeding 0.4, suggesting that moderate to slightly larger values are more conducive to performance. The algorithm parameter demonstrates minimal impact on accuracy with little difference in distribution.

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

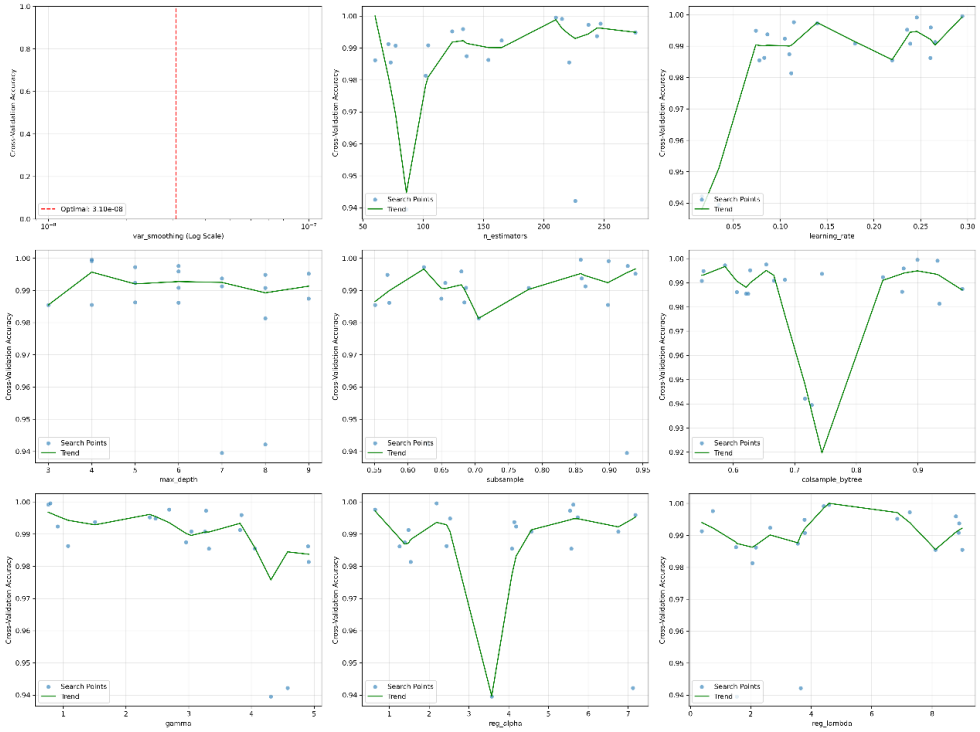

Figure 4: The influence of hyperparameters on Naive Bayes and XGBoost models.

The data in Fig.4 show that the var_smoothing parameter has a notable effect on accuracy for the NB model: as the parameter changes on a logarithmic scale, accuracy first increases and then decreases, indicating an optimal value where both excessively large and small values lead to accuracy decline.

In the XGBoost model, increasing n_estimators generally raises accuracy until it stabilizes; lower learning_rate values correspond to lower accuracy, while moderate values significantly improve and maintain high accuracy. The max_depth parameter shows relatively minor influence within the range of 2 to 6, and high and stable accuracy is observed when subsample approaches 1.0, colsample_bytree ranges between 0.7 and 0.9, gamma is relatively small, min_child_weight is 1–3, reg_alpha is 0–2, and reg_lambda is 2–6, indicating that appropriate parameter adjustment can enhance model performance.

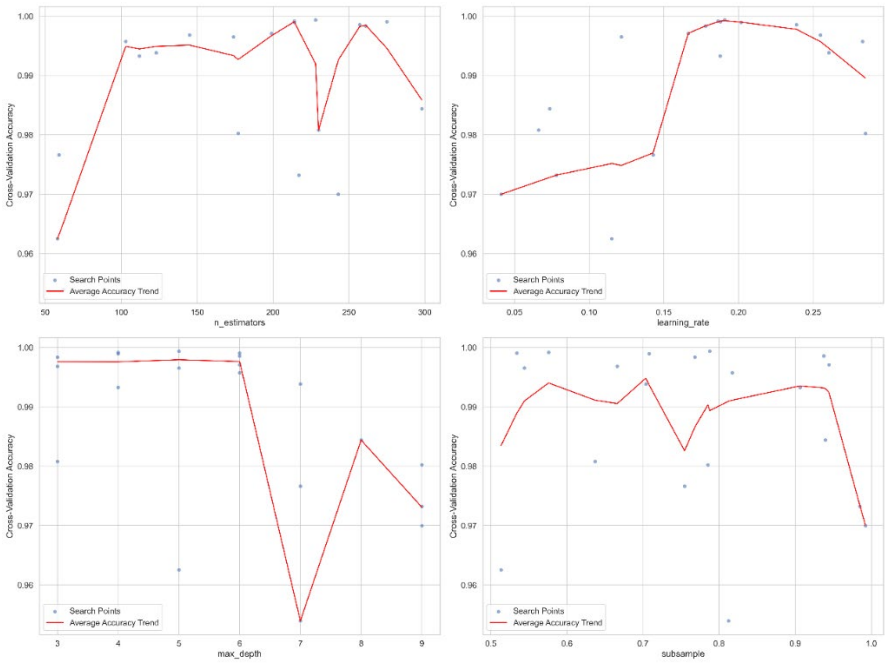

Figure 5: The influence of hyperparameters on the GBDT model.

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

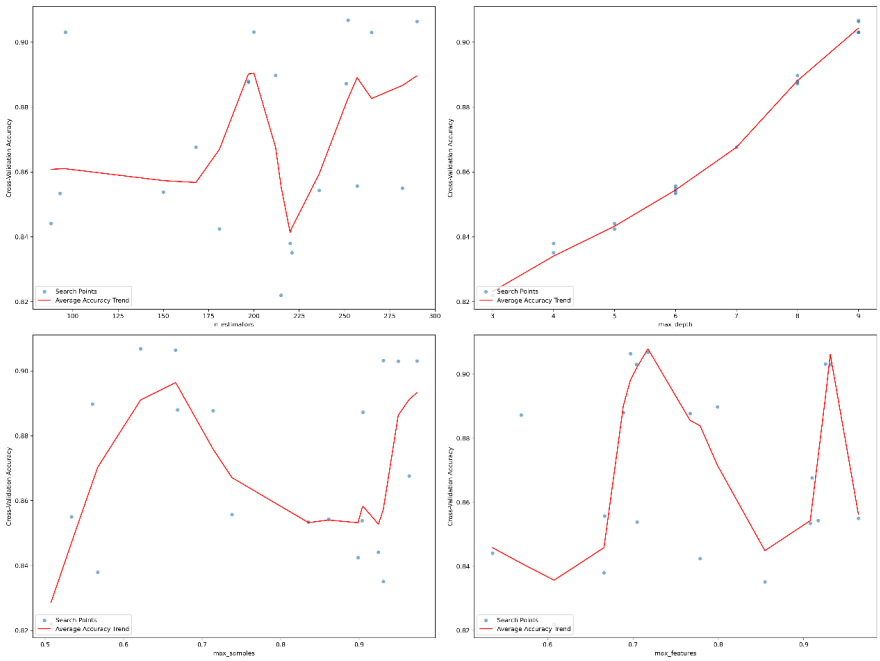

Figure 6: The influence of hyperparameters on the Random Forest model.

For GBDT, n_estimators boosts accuracy with fluctuations (50–300) before plateauing, while learning_rate peaks at 0.1–0.2, with higher values degrading performance. Excessive max_depth reduces accuracy, requiring depth constraints to curb overfitting. Subsample (0.5–1.0) introduces instability, necessitating careful calibration.

In RF, n_estimators (100–300) drives rapid accuracy gains followed by stabilization. Max_depth (3–9) enhances performance but risks overfitting at higher values. Max_samples (0.5–0.9) and min_samples_leaf variations introduce accuracy variability, demanding bias-variance balancing. It is not difficult to see in Fig.5 and Fig.6, both models require meticulous hyperparameter tuning to harmonize complexity and generalization.

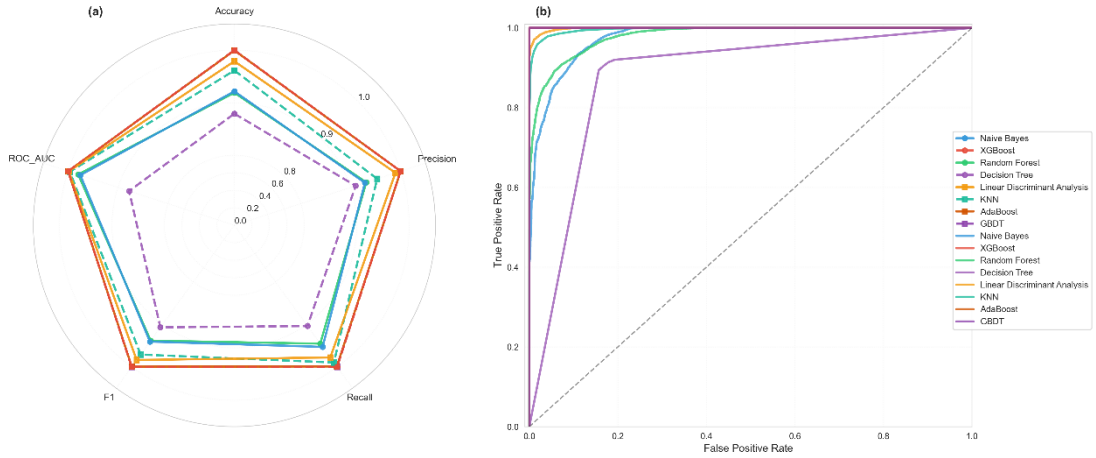

Figure 7: Radar chart and ROC curves of each model.

Fig.7 plots the model performance graph, from the radar chart, XGBoost, AdaBoost and GBDT all perform exceptionally well in the five indicators of Accuracy, Precision, Recall, F1-score and ROC_AUC, almost reaching the edge of the radar chart, with excellent and balanced comprehensive performance. Most indicators of linear discriminant analysis and KNN performed well, some indicators of Naive Bayes and random forest were acceptable, while the performance of each indicator of decision tree was relatively poor. In the ROC - AUC curve, the AUC values of XGBoost, AdaBoost and GBDT are 1.0000, which can perfectly distinguish positive and negative samples. The AUC value of linear discriminant analysis was 0.9987, and the KNN was 0.9965, indicating a strong classification ability. The AUC value of the random forest is 0.9798, and that of Naive Bayes is 0.9755. The performance is acceptable. The

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

AUC value of the decision tree is 0.8777, and its discrimination ability is relatively weak.

Table 1: This caption has one line so it is centered.

Model	Hyperparameter	Value/Type	Model	Hyperparameter	Value/Type
Naive Bayes	var_smoothing	1.0577e-09	KNN	n_neighbors	50
XGBoost	n_estimators	299		weights	uniform
	learning_rate	0.27721010		p	2
	max_depth	3		algorithm	ball_tree
	subsample	0.58266210		leaf_size	28
	colsample_bytree	0.56644768	Decision Tree	max_depth	15
	gamma	2.1428768579		min_split	2
	reg_alpha	0.0		min_ leaf	1
	reg_lambda	4.84974219		max_features	0.43758408
Random Forest	n_estimators	246	GBDT	n_estimators	243
	max_depth	10		learning_rate	0.3
	max_samples	0.5		max_depth	4
	max_features	0.78138232		subsample	0.70385756
AdaBoost	n_estimators	211	LDA	solver	lsqr
	learning_rate	0.77834859		shrinkage	0.05958398
	algorithm	SAMME.R		tol	0.001

Table 2: Performance Index Results of each model.

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Adaboost	1.0000	1.0000	1.0000	1.0000	1.0000
Random Forest	0.9223	0.9352	0.9459	0.9405	0.9802
Linear Discriminant Analysis	0.9946	0.9934	0.9983	0.9959	0.9999
Decision Tree	0.8791	0.9098	0.9038	0.9068	0.8777
KNN	0.9614	0.9529	0.9896	0.9709	0.9965
GBDT	0.9998	0.9997	1.0000	0.9998	1.0000
XGBoost	0.9999	0.9998	1.0000	0.9999	1.0000
Naive Bayes	0.9296	0.9359	0.9572	0.9464	0.9779

Table 1 shows the Settings of hyperparameters. And it can be obtained from the evaluation indicators in Table 2, Adaboost, GBDT, and XGBoost dominated classification metrics, achieving near-theoretical-maximum performance with exceptional discrimination of positive/negative samples. These models exhibited high precision in positive-class identification and balanced multi-class recognition. Linear Discriminant Analysis (LDA) followed closely but lagged slightly behind the top models. Decision Trees underperformed significantly across all metrics, showing limited sample discrimination and elevated misclassification rates. Random Forest and Naive Bayes delivered mid-tier results, while KNN demonstrated selective competency but remained inferior to the leading models. Quantitatively, Adaboost/GBDT/XGBoost are optimal for high-precision tasks, LDA serves as a competitive alternative, and Decision Trees necessitate architectural revisions or replacement in precision-critical applications.

5. Conclusion

5.1. Validation of the Effectiveness of Moss Growth Optimization Algorithm

The MGO algorithm simulates moss reproduction and cryptobiosis, combining spore diffusion for global hyperparameter exploration with dual reproduction for localized refinement. Experimental results reveal minor metric declines in select models but significant improvements in key metrics, particularly for ensemble learners. This demonstrates MGO's efficacy in optimizing hyperparameters to enhance model fitting and generalization, supported by empirical evidence across tested scenarios.

5.2. Research Value and Practical Significance

High-performance models like XGBoost and AdaBoost show strong potential for stroke risk assessment, supporting early screening and personalized interventions while reducing diagnostic errors. The MGO algorithm introduces a hyperparameter tuning framework for medical data, improving minority-class recognition in imbalanced datasets through dynamic search strategies. Current limitations

ISSN 2616-5775 Vol. 8, Issue 6: 36-44, DOI: 10.25236/AJCIS.2025.080605

include reliance on Kaggle data, requiring future validation with clinical datasets. Further work should integrate multimodal data, enhance MGO's computational efficiency, and explore hybrid methods (e.g., MGO-Bayesian optimization) to strengthen clinical risk assessment systems.

5.3. Conclusion Summary

This study optimized machine learning hyperparameters via the MGO algorithm, achieving significant results in stroke risk assessment: integrated models (XGBoost, AdaBoost, GBDT) achieved top performance, while MGO boosted adaptability to complex medical data. The framework provides methodological support for medical ML optimization and advances precision medicine in stroke prevention. Future work should expand datasets, enhance algorithm efficiency, and pursue cross-disciplinary integration to strengthen risk assessment robustness.

References

[1] Mukherjee, D., & Patil, C. G. (2011). Epidemiology and the global burden of stroke. World neurosurgery, 76(6), S85-S90.

[2] Boehme, A. K., Esenwa, C., & Elkind, M. S. (2017). Stroke risk factors, genetics, and prevention. Circulation research, 120(3), 472-495.

[3] Shehab, M., Abualigah, L., Shambour, Q., Abu-Hashem, M. A., Shambour, M. K. Y., Alsalibi, A. I., & Gandomi, A. H. (2022). Machine learning in medical applications: A review of state-of-the-art methods. Computers in Biology and Medicine, 145, 105458.

[4] Kaur, S., Aggarwal, H., & Rani, R. (2020). Hyper-parameter optimization of deep learning model for prediction of Parkinson's disease. Machine Vision and Applications, 31, 1-15.

[5] Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 17(1), 26-40.

[6] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia tools and applications, 80, 8091-8126.

[7] Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. Archives of computational methods in engineering, 29(5), 2531-2561.

[8] Ponce de León, I., & Montesano, M. (2017). Adaptation mechanisms in the evolution of moss defenses to microbes. Frontiers in Plant Science, 8, 366.

[9] Zheng, B., Chen, Y., Wang, C., Heidari, A. A., Liu, L., & Chen, H. (2024). The moss growth optimization (MGO): concepts and performance. Journal of Computational Design and Engineering, 11(5), 184-221.

[10] Jamthikar, A., Gupta, D., Khanna, N. N., Saba, L., Araki, T., Viskovic, K., ... & Suri, J. S. (2019). A low-cost machine learning-based cardiovascular/stroke risk assessment system: integration of conventional factors with image phenotypes. Cardiovascular diagnosis and therapy, 9(5), 420.

[11] Orfanoudaki, A., Chesley, E., Cadisch, C., Stein, B., Nouh, A., Alberts, M. J., & Bertsimas, D. (2020). Machine learning provides evidence that stroke risk is not linear: The non-linear Framingham stroke risk score. PloS one, 15(5), e0232414.

[12] Heo, J., Yoon, J. G., Park, H., Kim, Y. D., Nam, H. S., & Heo, J. H. (2019). Machine learning–based model for prediction of outcomes in acute stroke. Stroke, 50(5), 1263-1265.

[13] Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., ... & Warier, P. (2018). Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study. The Lancet, 392(10162), 2388-2396.

[14] Daidone, M., Ferrantelli, S., & Tuttolomondo, A. (2024). Machine learning applications in stroke medicine: advancements, challenges, and future prospectives. Neural regeneration research, 19(4), 769-773.

[15] Li, X., Bian, D., Yu, J., Li, M., & Zhao, D. (2019). Using machine learning models to improve stroke risk level classification methods of China national stroke screening. BMC medical informatics and decision making, 19, 1-7.

[16] Hassan, A., Gulzar Ahmad, S., Ullah Munir, E., Ali Khan, I., & Ramzan, N. (2024). Predictive modelling and identification of key risk factors for stroke using machine learning. Scientific Reports, 14(1), 11498.

[17] Dritsas, E., & Trigka, M. (2022). Stroke risk prediction with machine learning techniques. Sensors, 22(13), 4670.