International Journal of Engineering and Manufacturing, **2026, 3, 1-10**
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijem.2026.03.01

Comparative Evaluation of Evolutionary Hyperparameter Optimization for Gradient Boosting Ensembles in Clinical Multi-Class Classification

Fayzullo Nazarov
Faculty of Artificial Intelligence and Digital Technologies, Samarkand State University, Samarkand city, University Avenue 15, 14100, Uzbekistan
E-mail: fayzullo-samsu@mail.ru
ORCID iD: https://orcid.org/ 0000-0003-3925-406X

Shokhrukh Sariyev*
Faculty of Artificial Intelligence and Digital Technologies, Samarkand State University, Samarkand city, University Avenue 15, 14100, Uzbekistan
E-mail: sariyevshokhrukh@gmail.com
ORCID iD: https://orcid.org/ 0009-0002-3643-2027
*Corresponding author

Mekhriddin Nurmamatov
Faculty of Artificial Intelligence and Digital Technologies, Samarkand State University, Samarkand city, University Avenue 15, 14100, Uzbekistan
E-mail: mehriddinnur@gmail.com
ORCID iD: https://orcid.org/ 0000-0002-9619-5791

Islom Yalgoshev
Department Faculty of Artificial Intelligence and Digital Technologies, Samarkand State University, Samarkand city, University Avenue 15, 14100, Uzbekistan
E-mail: islomyalgoshev95@gmail.com
ORCID iD: https://orcid.org/ 0009-0006-0999-5472

Received: 28 August, 2025; Revised: 25 February, 2026; Accepted: 28 March, 2026; Published: 08 June, 2026

Abstract: In this study, the hyperparameters of Stochastic Gradient Boosting, LightGBM, and Histogram-based Gradient Boosting models were optimized using evolutionary algorithms. The main goal of this research is to find the most effective combination of hyperparameters for each model. Their goal is to increase accuracy and improve computational efficiency. The study was conducted on the basis of real clinical data. The data were obtained from the Samarkand City Endocrinology Center, using a dataset of diabetes-related and clinical indicators. The data were initially cleaned and processed using normalization, imputation, and 5-point cross-validation. We used five evolutionary strategies to fine-tune the main hyperparameters: Differential Evolution (DE), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Dynamic Ensemble with Hyperband (DEHB).In the optimization, the F1-macro was chosen as the main fitness function. This allows for a balanced accuracy assessment for multi-class classification. Across all three prototypes, evolutionary tuning consistently led to better results than Grid Search and Random Search. SGB+DE accuracy 0.9098, F1-macro 0.9096 gave the best results for the single model, whereas DEHB and CMA-ES worked better for LightGBM and HGB, respectively. The absolute increases above the unadjusted baselines ranged from 0.3 to 0.6 percentage points. They were small but could be repeated reliably across all evaluation criteria. During this study, the models were evaluated on the metrics of accuracy, precision, recall, and F1-macro.

Index Terms: Hyperparameter optimization, gradient boosting, ensemble learning, genetic algorithm, differential evolution, CMA-ES, DEHB, multi-class classification, clinical decision support

1. Introduction

One of the important areas of improving model efficiency used in machine learning systems today is the process of optimal selection of hyperparameters. The final result of any applied algorithm, i.e., prediction accuracy and generalization ability, largely depends on the correct selection of hyperparameters. In ensemble models based on gradient boosting, namely Stochastic Gradient Boosting, Light Gradient Boosting Machine, and Histogram-based Gradient Boosting, these parameters directly affect the learning rate, depth, number of leaves, regularization level, and number of trees of the model [1-4]. Therefore, optimizing hyperparameters in such models is one of the most urgent tasks from a scientific and practical point of view.

Traditional approaches to hyperparameter optimization include Grid Search and Random Search. These approaches are resource-intensive, but they are often ineffective in finding the global optimal solution. In this context, evolutionary algorithms are increasingly being used as an effective alternative to global optimization. Evolutionary approaches are based on the principles of natural selection, mutation, and adaptation, and provide stable results in complex, nonlinear, and multidimensional search spaces. Methods such as Differential Evolution and Genetic Algorithm are widely used in gradient-free optimization problems, and in addition to them, modern evolutionary approaches such as CMA-ES and Particle Swarm Optimization provide a balance between speed and accuracy in searching for hyperparameters [5-7]. The goal of this research is to optimize the hyperparameters of the StochasticGB, LightGBM, and HGB models using various evolutionary algorithms and find the combination with the highest accuracy among them. The research was based on data from diabetes patients from the Samarkand city Endocrinology center [8-9]. Performance was evaluated by 5-fold cross-validation (corrected from “5-phase”) utilizing accuracy, F1-macro, ROC-AUC, and log-loss as supplementary metrics. This study distinguishes itself from previous research through its extensive cross-evaluation: instead of coupling one optimizer with one model, we assess five evolutionary techniques against three different boosting architectures using the same clinical dataset under uniform settings. This methodology facilitates direct, unambiguous comparisons of optimizer-model compatibility a query that isolated research cannot resolve. A supplementary contribution is the clinical foundation: the dataset embodies authentic diabetes risk stratification from the Samarkand City Endocrinology Center, linking the methodological discoveries to a health-related categorization endeavor.

2. Dataset and Preprocessing

All classification studies utilized a balanced, structured health survey dataset specifically designed for multi-class diabetes risk prediction. There are 12066 entries in the whole corpus, divided into three target categories with 4022 instances each. Each category represents a clinically significant level of diabetes risk. Before modeling, a preprocessing pipeline cut down the initial pool of variables to thirteen characteristics that together show demographic background and self-reported clinical symptoms.

The following predictors were kept: Body Mass Index (TMI), Age, Sex, Self-Rated General Health on a five-category scale (USX1_5), Diagnosed High Blood Pressure (YQB), Physical Inactivity (KHTT), First-Degree Family History of Diabetes (OTDM), Polydipsia (HTCH), Polyuria (TTPCH), Impaired Wound Healing (YSB), Peripheral Numbness or Tingling (QAU), Visual Disturbance (KNP), and Mobility Limitation when Walking or Ascending Stairs (YYZCHQ). From a typing point of view, Sex is a dichotomous variable coded as 0 and 1, Self-Rated Health is an ordinal integer on the interval [1, 5] and the other eleven variables are binary flags that come from yes-or-no questions. Tree-based ensembles and attention-driven tabular designs do not require ordinal or binary inputs to be distributed in any way. This is why such variables were allowed into the pipeline without any further encoding or scaling. The only predictor with a continuous value, Body Mass Index, was put via min-max normalization mapping to fit values within the range [0, 1]. This was done to stop the raw magnitude from having too much of an effect on any distance-sensitive or gradient-sensitive part of the optimization process.

The data were split into three groups training (70%, n = 8445), validation (15%, n = 1811, which was only used for fitness evaluation during the search), and a held-out test set (15%, n = 1811, which was only used during the final assessment). We always used a random seed of 42, which means that all of the results we reported can be repeated exactly.All splitting and weight-initialization methods used the same global random seed of 42, which means that anyone can reproduce any reported result without needing to access the original raw data files[27].

StochasticGradientBoosting is a powerful ensemble method in machine learning that combines the results of decision trees from a large number of simple models to achieve high accuracy. In this model, only a small random portion of the training data is taken at each iteration and a new model is trained on this portion. The data set is entered in the following $\{a_j, b_j\}_{j=1}^{N}$ order, that is, the function of the initial model is expressed by formula (1).

$$F_0(a) = \arg\min_{\gamma} \sum_{j=1}^{N} \Upsilon(b_j, \gamma) \tag{1}$$

Multi-class classification problems are expressed using log-odds, and the loss function is calculated using categorical cross-entropy, as given in formula (2).

$$\Upsilon = -\sum_{j=1}^{N}\sum_{k=0}^{2} b_{j,k}\log(z_{k,j}) \tag{2}$$

Here $b_{j,k}, j-$ of the sample $k-$ class belonging indicator. $z_{j,k}-$ The probability predicted by the model is calculated by formula (3).

$$z_{j,k} = \frac{\exp(F_k(a_j))}{\sum_{i=0}^{2}\exp(F_i(a_j))} \tag{3}$$

At each stage, an iterative update of the model is performed when $m = 1,...,M$ is present. To reduce overfitting, *n*% samples are selected from the data set by random selection. The following formula (4) is used to calculate the model gradient.

$$h_{j,k}^{(m)} = \frac{\partial\Upsilon}{\partial F_k^{(m-1)}(a_j)} \tag{4}$$

Here

$F_k^{(m-1)}(a_j)-$ The model prediction from the previous step is expressed as

$z_{j,k}^{(m-1)}-$ Calculating probability using softmax.

After calculating the gradient, a new decision tree is built. The new tree $g_{m,k}(a)$, built using random samples and their gradients, is trained and used to reduce the errors in the previous model. The model is updated using formula (5).

$$F_k^{(m)}(a) = F_k^{(m-1)}(a) - \upsilon\cdot g_{m,k}(a) \tag{5}$$

(6) The final prediction result of the model is calculated by the formula.

$$\hat{b}(a) = \arg\max_k \frac{\exp(F_k^{(M)}(a))}{\sum_{j=0}^{K}\exp(F_i^{(M)}(a))} \tag{6}$$

The output classification result is used for each case.

LightGBM is a lightweight, fast, and efficient version of gradient boosting algorithms, optimized for large datasets. It works by sequentially building decision trees to create an ensemble of new models[13-14]. We can give a general view of the LightGBM ensemble model expressed by formula (7).

$$F_k(a_j) = \sum_{m=1}^{M} f_{m,k}(a_j) \tag{7}$$

Here

$F_k(a_j)$, $k-$ Generalized prediction for a given class;

$f_{m,k}(a_j)$, $k-$ decision tree prediction for a given class;

The probability of each class is calculated using formula (8).

$$z_k(a) = \frac{e^{F_k^{(a)}}}{\sum_{i=0}^{K} e^{F_i^{(a)}}} \tag{8}$$

The final class is determined using formula (9).

$$\hat{b}(a) = \arg\max_{k} z_k(a) \tag{9}$$

(10) With the formula, we can calculate the loss function of a multi-class classification.

$$\Upsilon = -\sum_{j=1}^{N}\sum_{k=0}^{K} b_{j,k}\log(z_{j,k}) \tag{10}$$

To build the next tree, the loss function is used to calculate the gradient and Hessian values. The gradients are calculated using formula (11).

$$h_{j,k}^{(m)} = \frac{\partial \Upsilon}{\partial F_k(a_j)} \tag{11}$$

In multi-class classification models, for each sample taken, the output logit for each class is generated and converted into probabilities using the softmax function. For each class, the model calculates a value in the range [0-1], and they sum to 1 across all classes. The one-hot representation gives the true values. The correct class component is 1 and the rest are 0. The gradient of the cross-entropy loss with respect to the logit is the difference between the model's output probability and the true one-hot probability, and is used to measure the model's error. The second-order Hessian is used for node and leaf counts. In softmax cross-entropy, the second derivative on the diagonal represents the probability of each class. The second derivative decreases as the probability approaches 0 or 1 [15]. It shows a larger result for average values. In most implementations, only the diagonal part of the Hessian is used, which greatly simplifies the calculation. To build a tree, gradients and Hessians are collected across samples. L2-Leaf weights are calculated taking regularization into account. The optimization is based on Newton's approximation, where the loss at each $m-$ iteration is calculated using the gradient over Υ and the Hessian second-order derivative formula (12).

$$g_{j,k}^{(m)} = \frac{\partial^2 \Upsilon}{\partial (F_k(a_j))^2} \tag{12}$$

$g_{j,k}^{(m)}-$ gives the local curvature, i.e. second-order information. The weight of each leaf is expressed by the following formula (13).

$$\omega^{*} = -\frac{\sum_{j\in leaf} h_{j,k}^{(m)}}{\sum_{j\in leaf} g_{j,k}^{(m)} + \lambda} \tag{13}$$

In the next step, the Gain function (14) below is used to select the optimal split during the tree construction process.

$$Gain = \frac{1}{2}\left[\frac{H_{\Upsilon}^2}{G_{\Upsilon}+\lambda} + \frac{H_R^2}{G_R+\lambda} - \frac{(H_{\Upsilon}+H_R)^2}{G_{\Upsilon}+G_R+\lambda}\right] - \gamma \tag{14}$$

Here $H_{\Upsilon} = \sum_{j\in\Upsilon} g_j$, $H_R = \sum_{j\in R} h_j$ represents the sum of the gradients in the left and right sections.

$G_{\Upsilon} = \sum_{j\in\Upsilon} g_j$, $G_R = \sum_{j\in R} g_j$ The sum of the Hessians in the left and right sections. λ is the "Penalty" parameter used to add a new node. The split with the highest gain value in the model is selected. The process of updating the model iterations is carried out in the following sequence. After adding a new tree, the model is updated according to the following formula (15).

$$F_k^{(m)}(a) = F_k^{(m-1)}(a) + \upsilon \cdot f_{m,k}(a) \tag{15}$$

After completing all the above steps, the final appearance of the model will be as shown in (16) below.

$$\hat{b}(a) = \arg\max_{k\in\{0,,1,2\}} \left(\frac{e^{\sum_{m=1}^{M} f_{m,k}(a)}}{\sum_{j=0}^{2} e^{\sum_{m=1}^{M} f_{m,j}(a)}} \right) \tag{16}$$

The HistGradientBoosting Classifier ensemble is an optimized version of the Gradient Boosting Decision Tree algorithm, which divides the entire range of values for each feature into pre-defined small bin-histogram segments and searches for splits exactly along these histogram bins. The model views each feature as a discrete histogram bin instead of the exact values of each feature [17-20]. The process of finding the optimal point for the split is faster and leads to reduced memory consumption. This approach allows for efficient computation even with millions of samples and many features. By dividing the HistGB ensemble into bins, it speeds up the processing of the dataset and is more efficient than other ensemble models when working with large datasets. The discretization of the dataset into bins is performed using the following formula (17). For each feature $M_i \in \{1,2,...,M\}$, the range of values is divided into D bins.

$$\begin{cases} bin_i = \{d_0, d_1, ..., d_D\} \\ d_0 = \min\limits_j a_j^i \\ d_D = \max\limits_j a_j^i \end{cases} \tag{17}$$

Each value is as follows $d_j^i = \max\{m : d_m \le a_j^i\}$, $m = 1,2,...,D$ bin index is converted.

In multi-class classification problems, the task is to classify each given sample into one of a certain class. Here, logit and softmax functions are used to correctly estimate class probabilities and make predictions based on them. The probabilities and logit values for the initial classes are calculated using the formula (18) for the probability of samples for each class $k \in \{1,2,...,K\}$ in the data set.

$$h_k = \frac{1}{N}\sum_{j=1}^{N} I(b_j = k) \tag{18}$$

The indicator function is expressed in the form of formula (19)

$$I = \begin{cases} if\ b_j = k\ be\ 1 \\ else\ 0 \end{cases} \tag{19}$$

For classification algorithms, the class probabilities are expressed in logarithmic form using the formula (20).

$$F_k^{(0)} = \ln(h_k) \tag{20}$$

To obtain class probabilities with the softmax function, normalization is required using the following formula (21).

$$\hat{z}_k^{(0)}(a_i) = \frac{e^{F_k^{(0)}}}{\sum_{l=1}^{K} e^{F_l^{(0)}}} \tag{21}$$

The number of iterations to update is $\tau = 1,2,...,\mathrm{T}$. The softmax probabilities are calculated by the formula (22).

$$\hat{z}_k^{(\tau-1)}(a_j) = \frac{e^{F_k^{(\tau-1)}(a_j)}}{\sum_{l=1}^{K} e^{F_l^{(\tau-1)}(a_j)}} \tag{22}$$

The calculation of the gradient and Hessian is carried out using the given formula (23).

$$\begin{cases} h_{j,k}^{(\tau)} = \hat{z}_k^{(\tau-1)}(a_j) - I\left(b_j = k\right) \\ g_{j,k}^{(\tau)} = \hat{z}_k^{(\tau-1)}(a_j)\left(1 - \hat{z}_k^{(\tau-1)}(a_j)\right) \end{cases} \tag{23}$$

$$\begin{cases} H_{k,i,q} = \sum\limits_{\{q_j^i = q\}} h_{j,k}^{(\tau)} \\ G_{k,i,q} = \sum\limits_{\{q_j^i = q\}} g_{j,k}^{(\tau)} \end{cases} \tag{24}$$

When finding the optimal split, the split for each q bin is optimized by formula (25).

$$Gain_{k,i,q} = \frac{H_Y^2}{G_Y + \lambda} + \frac{H_R^2}{G_R + \lambda} - \frac{(H_Y + H_R)^2}{G_Y + G_R + \lambda} - \gamma \tag{25}$$

The output value for each leaf is calculated by formula (26).

$$f_k^{(\tau)}(a_j) = -\frac{H_q}{G_q + \lambda} \tag{26}$$

Here $H_q, G_q -$ represents the gradient and Hessian sums in the leaves.

The model is updated using formula (27).

$$F_k^{(\tau)}(a_j) = F_k^{(\tau-1)}(a_j) + \vartheta f_k^{(\tau)}(a_j) \tag{27}$$

$$F_k^{(T)}(a_j) = F_k^{(0)} + \sum_{\tau=1}^{T} \vartheta f_k^{(\tau)}(a_j) \tag{28}$$

The following equation (29) represents the classification stage of the multi-class HistGradientBoosting model. The final logits obtained after ta iterations are normalized to probabilities using $F_k^{(T)}(a_j)$ softmax, and the class with the highest probability is identified.

$$\hat{z}_k^T(a_j) = \frac{e^{F_k^{(T)}(a_j)}}{\sum_{i=1}^{K} e^{F_i^T(a_j)}} \quad \hat{b}_j = \arg\max_k \hat{z}_k^{(T)}(a_j) \tag{29}$$

3. Optimization

The hyperparameter optimization task is expressed as follows (30).

$$\Theta^* = \arg\max_{\Theta \in \Omega} F_{F1}(\Theta) \tag{30}$$

Here $\Theta = \{o_1, o_2, ..., o_n\}$ set of model hyperparameters, $\Omega -$ parameters search space, $F_{F1}(\Theta)$, F1-macro estimation function calculated based on 5-fold cross-validation. This objective function is iteratively optimized using evolutionary algorithms. At each iteration, the new parameter combination is updated as follows[21-24].

$$X_i^{(t+1)} = X_i^{(t)} + \Delta_i^{(t)} \tag{31}$$

Here $\Delta_i^{(t)} -$ The direction of mutation or search depends on the type of algorithm. At each step, the fitness function is minimized as follows (32).

$$f(\Theta) = 1 - F_{F1}(\Theta) \tag{32}$$

As a result, the set of parameters Θ^* with the highest accuracy is determined as follows (33).

$$\Theta^* = \arg\min_{\Theta \in \Omega} f(\Theta) \tag{33}$$

This model represents the mathematical basis for the process of optimizing hyperparameters using evolutionary algorithms in SGB, LightGBM, and HGB models[25-26].

The working scheme of the genetic algorithm for ensemble models proposed in the study is presented in Figure 1. In the first stage of the algorithm, a random initial set of hyperparameters $P_0 = \{\Theta_1, \Theta_2, ..., \Theta_N\}$ is generated. The fitness function for each model is estimated by computing a multi-class cross-entropy loss function. In the next stage, the best

solutions are stored using an elitism strategy, and parents are selected through tournament selection. A new generation is generated using the crossover operator and mutation. The fitness values for the SGB, HistGradientBoosting, and LightGBM models are evaluated in parallel. The iterations continue until a stopping condition is met, and each model is retrained based on the final optimal hyperparameters.

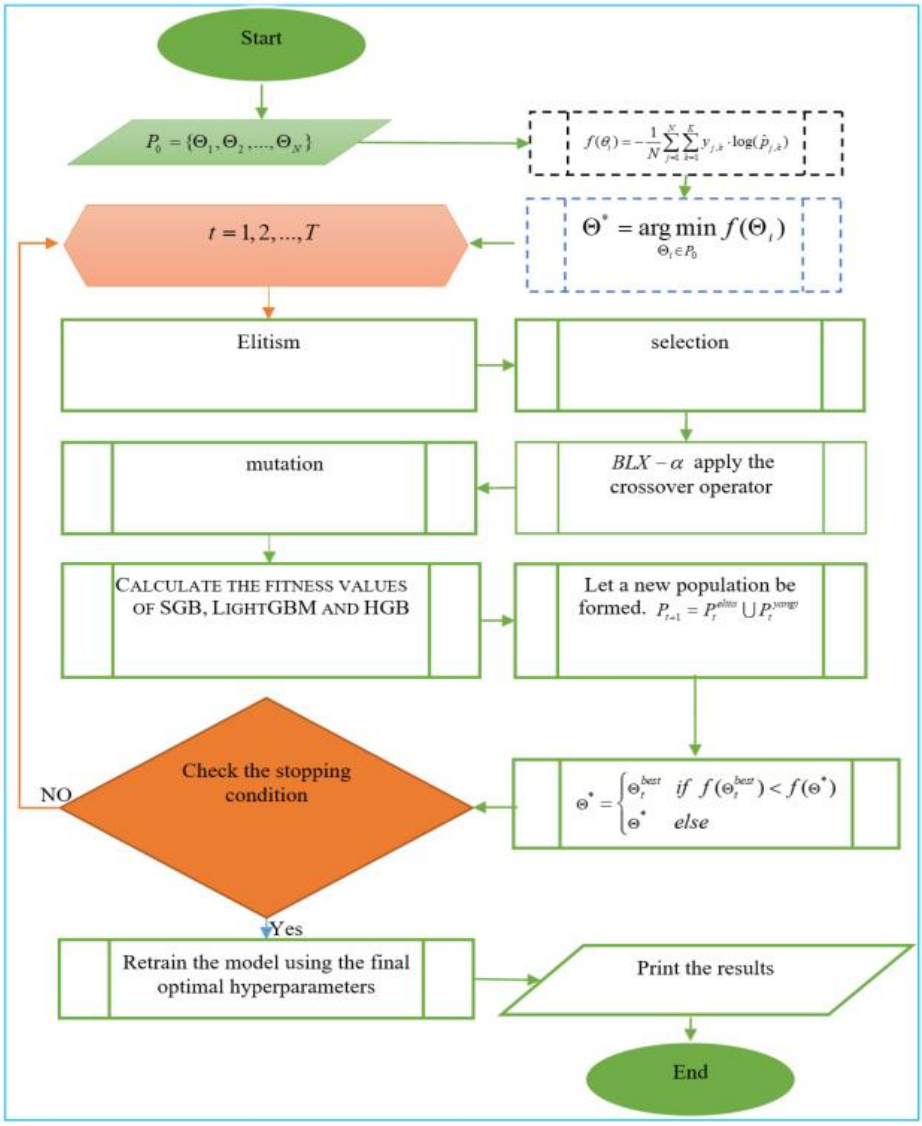

Fig. 1. Flow chart of Quantum Challenge Generation

4. Results

In the proposed authentication system, the Authentication Server initiates the process by generating a quantum challenge. This Challenge involves preparing a superposition of quantum states, which are key to leveraging the fundamentals of quantum physics for secure authentication [1].

Table 1. Ensemble Models of Machine Learning

Models	**Evaluation methods**			
SGB	Accuracy	Precision	Recall	F1-score
	0.9035	0.9134	0.9035	0.9021
LightGBM	0.9053	0.9152	0.9053	0.9040
HGB	0.9071	0.9175	0.9071	0.9058

The results of this study are presented in Table 1 below, with the values obtained without adjusting the hyperparameters. The results of the research conducted through the experimental results during this study show that the

results of optimization based on the Random Search and Grid Search algorithms, which are among the traditional optimization methods, were achieved as shown in Table 2 below.

Table 2. Ensemble Optimization with RS And GS Ensemble Machine Learning Models

Models	**Evaluation methods**			
SGB+RS	Accuracy	Precision	Recall	F1-score
	0.9041	0.9144	0.9043	0.9041
LightGBM+RS	0.9061	0.9162	0.9061	0.9060
HGB+RS	0.9081	0.9182	0.9080	0.9067
SGB+GS	0.9043	0.9144	0.9046	0.9031
LightGBM+GS	0.9063	0.9152	0.9062	0.9051
HGB+GS	0.9081	0.9174	0.9082	0.9069

Experimental results with the evolutionary optimization algorithms proposed in this study were obtained for each model. The experimental results obtained in this research work are presented in Table 3. From the results, we can see that evolutionary algorithms are better than traditional methods.

Table 3. Results Experimental Results Obtained in Research Work

Models	**Evaluation methods**			
SGB + DE	Accuracy	Precision	Recall	F1-score
	0.9098	0.9197	0.9096	0.9096
LightGBM + DEHB	0.9090	0.9105	0.9101	0.9094
HGB + CMA-ES	0.9096	0.9100	0.9095	0.9094

A comparative summary of the results of this study is presented in Fig. 2 below. In conclusion, it can be said that it is possible to apply the results of the optimization model through evolutionary algorithms, as can be seen from the experimental results.

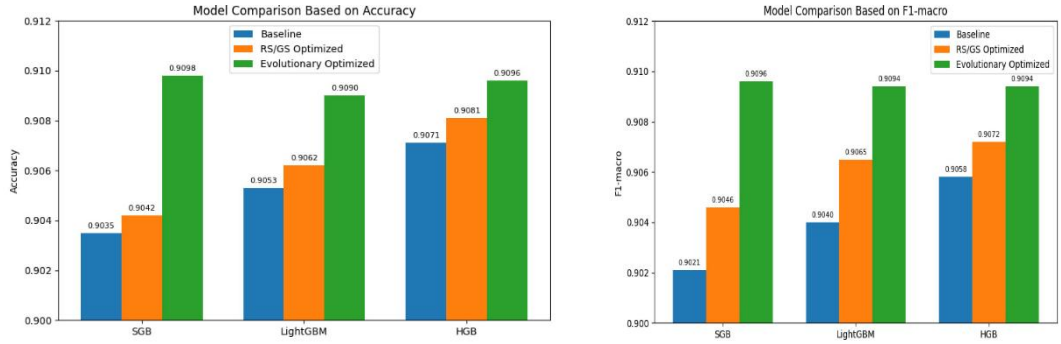

Fig.2. a)Comparison of accuracy of model b) F1-macro comparison of models

Figure 2 of the results of the study shows a comparison of the models in terms of accuracy and F1-macro evaluation criteria. The results show that all models have improved efficiency through optimization. In particular, the genetic algorithm optimization method has achieved higher results compared to the baseline and RS/GS optimized cases. The highest accuracy result was achieved by the SGB model, which was 0.9098, and by the F1-macro, which was 0.9096. In addition, it can be seen that the results of the HistGB and LightGBM models also improved steadily after evolutionary optimization. The results do not indicate a single generally superior optimizer, DE was optimal for SGB, DEHB for

LightGBM, and CMA-ES for HGB. This optimizer–model specificity means that the evolutionary strategy should be chosen based on the target architecture, not used the same way for all of them. GA did well for SGB, but it didn't do as well as LightGBM or HGB.

5. Conclusion

In this study, the model hyperparameters of LightGBM, SGB, and HGB were tested under various optimizations. The study compared and analyzed their accuracy indicators. Initial experiments showed that without hyperparameter tuning, the accuracy between models ranged from 0.9035-0.9071, and the F1-score ranged from 0.9021-0.9058, limiting the overall performance of the model. The results were slightly improved by optimizing the parameters using traditional methods, Grid Search and Random Search. These methods improved the models by 0.2-0.3%. It ensured the best accuracy in the HGB and LightGBM models. However, since traditional search methods are computationally intensive and limited in finding a global optimal solution, more advanced approaches were required. The results of optimization using evolutionary algorithms GA, DE, DEHB, and CMA-ES yielded stable and higher performance compared to traditional methods. Within the scope of this study three gradient boosting classifiers, five evolutionary optimizers, and one balanced clinical dataset the results confirm that evolutionary hyperparameter search yields consistent, repeatable accuracy gains over conventional Grid Search and Random Search methods. The improvements are modest in absolute terms (0.31-0.61%), yet they are stable across models and metrics. Whether these findings extend to imbalanced, larger, or differently structured datasets remains an open question that warrants further investigation.

References

[1] Z.He, D. Lin, T. Lau, and M. Wu, "Gradient Boosting Machine: A Survey," arXiv preprint arXiv:1908.06951, 2019.

[2] O. Bakhteev, "Comprehensive Analysis of Gradient-Based Hyperparameter Optimization," Machine Learning Research Reports, 2017.

[3] C. Meaney, X. Wang, J. Guan, and T. A. Stukel, "Comparison of Methods for Tuning Machine Learning Model Hyper-Parameters: With Application to Predicting High-Need High-Cost Health Care Users," BMC Medical Research Methodology, vol. 25, no. 3, pp. 112–128, 2025, doi: 10.1186/s12874-024-02389-5.

[4] A. A. Rustamovich and N. F. Makhmadiyarovich, "Development of Control Models of Non-Stationary Systems in Limited Conditions," 2020 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2020, pp. 1-4, doi: 10.1109/ICISCT50599.2020.9351372

[5] A. Axatov, M. Nurmamatov, F. Nazarov, and Sh. Sariyev, "Genetic Algorithm Application Technology in Multi-Parameter Optimization Problems," AIP Conf. Proc., vol. 3244, art. no. 030025, 2024, doi: 10.1063/5.0242074.

[6] M. Molina Van den Bosch, C. Montalbano, R. Dornberger, and T. Hanne, "Evolutionary, Bayesian, and Quasi-Monte Carlo Hyperparameter Tuning," in Advances in Computational Intelligence, Springer, 2025, pp. 512–524.

[7] N. Fayzullo, S. Sariyev and Y. Sherzodjon, "Analyzing the Effectiveness of Ensemble Methods in Solving Multi-Class Classification Problems," 2025 International Russian Smart Industry Conference (SmartIndustryCon), Sochi, Russian Federation, 2025, pp. 788-793, doi: 10.1109/SmartIndustryCon65166.2025.10986248.

[8] S. Sariyev, I. Yalgoshev, and M. Nuriddinova, "Traditional Methods and Modern Approaches Based on Ensemble Algorithms for Decision-Making in Diagnostics Using Medical Data," in Proc. 2025 Int. Russian Automation Conf. (RusAutoCon), Sochi, Russia, 2025, pp. 971–975, doi: 10.1109/RusAutoCon65989.2025.11177423.

[9] X. Liu and M. Yang, "Simultaneous Curve Registration and Clustering for Functional Data," Comput. Statist. Data Anal., vol. 53, no. 4, pp. 1361–1376, 2019.

[10] L. Ye and E. Keogh, "Time Series Shapelets: A New Primitive for Data Mining," in Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2009, pp. 947–956.

[11] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proc. IEEE Int. Conf. Neural Networks (ICNN), Perth, Australia, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[12] N. Awad, N. Mallik, and F. Hutter, "DEHB: Evolutionary Hyper-parameter Optimization at Scale," in Proc. 30th Int. Joint Conf. Artificial Intelligence (IJCAI), 2021, pp. 2147–2153, doi: 10.24963/ijcai.2021/296.

[13] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-Based Hyperparameter Optimization Through Reversible Learning," in Proc. 32nd Int. Conf. Machine Learning (ICML), vol. 37, 2015, pp. 2113–2122.

[14] H. Yang, G. Qin, Z. Liu, Y. Hu, and Q. Dai, "LightGBM Robust Optimization Algorithm Based on Topological Data Analysis," arXiv preprint arXiv:2406.13300, 2024.

[15] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," Proc. IEEE, vol. 104, no. 1, pp. 148–175, 2016, doi: 10.1109/JPROC.2015.2494218.

[16] N. Hansen and A. Ostermeier, "Completely Derandomized Self-Adaptation in Evolution Strategies," Evolutionary Computation, vol. 9, no. 2, pp. 159–195, 2001, doi: 10.1162/106365601750190398.

[17] T. Zhang and Y. Zhao, "Hyper-Parameter Tuned LightGBM Using Memetic Firefly Algorithm," Appl. Soft Comput., vol. 110, p. 107698, 2021, doi: 10.1016/j.asoc.2021.107698.

[18] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[19] S. Akhatkulov, I. Yalgoshev, and J. Haydarov, "Different Warmup and Annealing Strategies for ANN Models to Predict Air Quality Index," in Proc. 2025 Int. Russian Automation Conf. (RusAutoCon), Sochi, Russia, 2025, pp. 491–496, doi: 10.1109/RusAutoCon65989.2025.11177428.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 3146–3154.
[21] L. Liu and Y. Wang, "Evolutionary Differential Algorithm for Hyperparameter Optimization in Machine Learning Models," Expert Syst. Appl., vol. 223, p. 120902, 2023.
[22] D. Adhikari et al., "Improving Medical Data Classification with Hybrid Evolutionary LightGBM Models," Comput. Biol. Med., vol. 148, p. 105833, 2022.
[23] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 4–31, 2011, doi: 10.1109/TEVC.2010.2059031.
[24] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer, 2009, doi: 10.1007/978-0-387-84858-7.
[25] F. Hutter, L. Kotthoff, and J. Vanschoren, Automated Machine Learning: Methods, Systems, Challenges. Springer, 2019, doi: 10.1007/978-3-030-05318-5.
[26] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, 2nd ed. Springer, 2015, doi: 10.1007/978-3-662-44874-8.
[27] M. Feurer and F. Hutter, "Hyperparameter Optimization," in Automated Machine Learning: Methods, Systems, Challenges, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Springer, 2019, pp. 3–33, doi: 10.1007/978-3-030-05318-5_1.

Authors' Profiles

Dr. Nazarov Fayzullo Makhmadiyarovich holds the academic degree of Doctor of Sciences (DSc) in Technical Sciencesand the academic title of Associate Professor. Currently, Nazarov Fayzullo Makhmadiyarovich is the Dean of the Faculty of Artificial Intelligence and Digital Technologies of Samarkand State University. His research interestsinclude problems related to parallel computing, blockchain technologies, distributed systems, and deep learning.

Shokhrukh Sariyev is currently a PhD researcher at Samarkand State University, Samarkand, Uzbekistan. His research interests include machine learning, artificial intelligence, deep learning and intelligent medical diagnostic systems.

Dr. Mehriddin Q. Nurmamatov obtained his PhD in Technical Sciences and holds the academic title of Associate Professor. He is currently working as the Head of the Department of Control Theory and Information Security at Samarkand State University, Samarkand, Uzbekistan.

Islom Yalgoshev is currently a PhD student in the Faculty of Artificial Intelligence and Digital Technologies, Samarkand State University, Samarkand, Uzbekistan. His research interests include machine learning, hyperparameter optimization, ensemble learning methods.

How to cite this paper

Fayzullo Nazarov, Shokhrukh Sariyev, Mekhriddin Nurmamatov, Islom Yalgoshev, "Comparative Evaluation of Evolutionary Hyperparameter Optimization for Gradient Boosting Ensembles in Clinical Multi-Class Classification", International Journal of Engineering and Manufacturing (IJEM), Vol.16, No.3, pp.1-10, 2026. DOI:10.5815/ijem.2026.03.01