Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 31s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Hyperparameter-Tuned Machine Learning Model for Accurate Prediction of Heart Disease

Dr. B. Suresh Kumar[1], Dr. L. Maria Anthony Kumar*[2]

[1]Assistant Professor, Department of Computer Science, Government Arts and Science College, Sriperumbudur, Kundrathur, Chennai-600069. (Deputed from Annamalai University).

Email ID: sureshaucis@gmail.com

[2*]Assistant Professor, Department of Computer Applications, Periyar Arts College, Cuddalore - 607001. (Deputed from Annamalai University).

Email ID: lmakumarphd@gmail.com

***Corresponding Author**

Dr. L. Maria Anthony Kumar,

[2]Assistant Professor, Department of Computer Applications, Periyar Arts College, Cuddalore - 607001. (Deputed from Annamalai University).

Email ID: lmakumarphd@gmail.com

Cite this paper as: Dr. B. Suresh Kumar, Dr. L. Maria Anthony Kumar, (2025) Hyperparameter-Tuned Machine Learning Model for Accurate Prediction of Heart Disease, *Journal of Neonatal Surgery*, 14 (31s), 135-143

ABSTRACT

Cardiovascular disease (CVD) continues to be a major global health concern, contributing significantly to morbidity and mortality rates. Early and accurate diagnosis of heart disease is crucial for timely intervention and improved patient outcomes. In this study, we present a robust machine learning framework enhanced through systematic hyperparameter tuning for the identification of heart disease. The well-known Cleveland Heart Disease dataset from the UCI Machine Learning Repository is employed as the primary dataset for model development and evaluation. The proposed methodology begins with the preprocessing of the dataset, followed by the extraction of relevant features essential for classification. These features are then supplied to multiple machine learning classifiers, where the performance of each model is refined using advanced hyperparameter tuning techniques. Specifically, four prominent tuning strategies are explored: Grid Search, Random Search, Halving Grid Search, and Halving Random Search. These techniques are applied to optimize the hyperparameters of various classifiers, with the objective of maximizing prediction accuracy.

Through extensive experimentation, the Random Forest classifier optimized via Random Search emerged as the most effective model, achieving an impressive accuracy of **92.45%** in detecting heart disease. This significant result highlights the impact of appropriate hyperparameter tuning on the performance of machine learning algorithms, particularly in medical data classification tasks. The findings of this study demonstrate that incorporating systematic hyperparameter optimization into the machine learning pipeline not only enhances diagnostic accuracy but also improves the generalizability and reliability of predictive models in healthcare. The proposed framework shows promise as a decision-support tool that can aid medical professionals in the early and accurate detection of cardiovascular diseases

Keywords: *Cardiovascular Disease, Hyper parameters, Machine learning algorithms, Cleveland dataset, Grid Search, Random Search, Halving Grid Search, and Halving Random Search.*

1. INTRODUCTION

Non-communicable diseases (NCDs) account for nearly 70% of all deaths globally, with cardiovascular diseases (CVDs) being one of the most prominent contributors, alongside conditions such as stroke, cancer, diabetes, and chronic respiratory diseases [1]. Often progressing without overt symptoms, CVDs silently affect the heart and blood vessels, encompassing a range of conditions including coronary artery disease, cerebrovascular disease, rheumatic heart disease, and others. Traditionally, the diagnosis of such diseases relies on routine medical evaluations and clinical expertise. However, timely and accurate detection remains a significant challenge due to the complex and multifactorial nature of these conditions. This has driven considerable research interest in the development of computational models, particularly those based on machine learning (ML), to assist in the early prediction and diagnosis of cardiovascular diseases [2][3]. In the context of machine learning, parameters are internal to the model and are learned from the data during training. In contrast, hyperparameters are

external configuration settings that govern the learning process itself such as the number of trees in a random forest, learning rate in gradient boosting, or kernel type in support vector machines. These values are not learned from the data but must be specified prior to training. The performance of an ML model is significantly influenced by the selection of appropriate hyperparameter values. Thus, hyperparameter tuning is the process of systematically identifying the most effective combination of hyperparameter values to maximize model performance within a reasonable computational time [4].

In recent years, machine learning has revolutionized healthcare diagnostics by enabling automated systems to recognize patterns in complex datasets, offering promising results in areas such as radiology, oncology, and cardiology. Specifically, in cardiovascular disease prediction, ML models have shown the ability to outperform traditional statistical techniques by capturing nonlinear relationships and interactions between clinical features. However, the accuracy and reliability of these models are heavily dependent on how well they are configured. Without proper tuning, even powerful algorithms like Random Forest, XGBoost, or Support Vector Machines may fail to deliver optimal results, leading to underperformance or misleading predictions that can have critical consequences in a clinical setting [5]. Hyperparameter tuning addresses several key challenges in CVD prediction models, including overfitting, underfitting, and model generalization. Manual tuning, while intuitive, becomes impractical as the dimensionality of hyperparameter space increases. Therefore, automated methods such as Grid Search, Random Search, Halving Grid Search, Bayesian Optimization, and others are employed to systematically explore and evaluate combinations of hyperparameters. These methods help identify the configuration that maximizes accuracy while maintaining computational efficiency. Moreover, some techniques, such as Bayesian Optimization and Tree-structured Parzen Estimators (TPE), leverage probabilistic models to intelligently guide the search, making them more efficient than exhaustive search methods [6][7]. In healthcare applications where accuracy, speed, and interpretability are critical, such optimization techniques are not only beneficial but essential. While many machine learning algorithms come with default hyperparameter values, these defaults do not guarantee optimal performance across diverse datasets or problem domains. Therefore, tuning hyperparameters is essential, particularly in sensitive applications like medical diagnosis, where accuracy can directly impact patient outcomes. Effective hyperparameter tuning not only improves predictive accuracy but also enhances the robustness and generalization capabilities of the model. This underscores the importance of incorporating tuning mechanisms into the design and deployment of ML models for cardiovascular disease detection and similar healthcare challenges

2. RELATED WORKS

Jinny et al. [8] developed a hybrid prediction model that integrates genetic algorithms (GAs), hyperparameter tuning, and various machine learning methods. Their model, tested on coronary heart disease data, demonstrated notable improvements in accuracy by leveraging evolutionary search techniques to optimize model parameters. Similarly, Asif et al. [9] performed a detailed comparative analysis of popular machine learning algorithms including logistic regression, decision trees, SVMs, and random forests evaluating them on metrics such as accuracy, precision, recall, and F1-score. Their findings underscored that model performance varies significantly depending on the choice of algorithm and feature selection strategy, highlighting the need for model-specific tuning. Hashi and Zaman [10, 11] proposed a structured machine learning pipeline that emphasized the importance of hyperparameter tuning in boosting the classification performance of predictive models. They showed that tuning parameters such as learning rate, tree depth, and kernel functions significantly impacts the predictive power of algorithms like random forest and SVM. Firdaus et al. [12] extended this work by applying deep neural networks (DNNs) combined with hyperparameter optimization, illustrating how deep learning models, when properly tuned, outperform traditional algorithms in detecting complex patterns associated with heart disease. Their use of advanced architectures and tuning strategies demonstrated the growing relevance of deep learning in medical analytics.

Further studies introduced optimization methods that enhance ensemble learning. Sonth et al. [13] optimized the random forest algorithm using a combination of ensemble techniques and hyperparameter tuning. Their study concluded that tuning ensemble models can mitigate issues like overfitting and bias, particularly in datasets with class imbalance or noise. Valarmathi and Sheela [14] proposed a systematic framework for heart disease prediction that incorporates Hyperparameter Optimization (HPO) strategies like grid search and random search. They emphasized that careful calibration of model parameters can yield significant performance gains, especially when combined with proper feature engineering. El-Shafiey et al. [15] introduced a hybrid approach involving Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) to optimize the random forest model. Their work demonstrated that combining metaheuristic optimization techniques can effectively navigate complex hyperparameter spaces and yield high-accuracy models. Finally, Bergstra et al. [16] provided a foundational study on scalable hyperparameter optimization methods, introducing random search and demonstrating its superiority over grid search in high-dimensional settings. Their work laid the groundwork for many of the hyperparameter tuning techniques applied in later medical and healthcare prediction systems.

3. PROPOSED METHODOLOGY

The proposed study comprises three major phases: data preprocessing, classification using hyperparameter-tuned machine learning models, and performance evaluation. These phases are designed to systematically process the dataset, build optimized predictive models, and rigorously assess their effectiveness in identifying heart disease. Figure 1 depicts the

various stages of the proposed model.

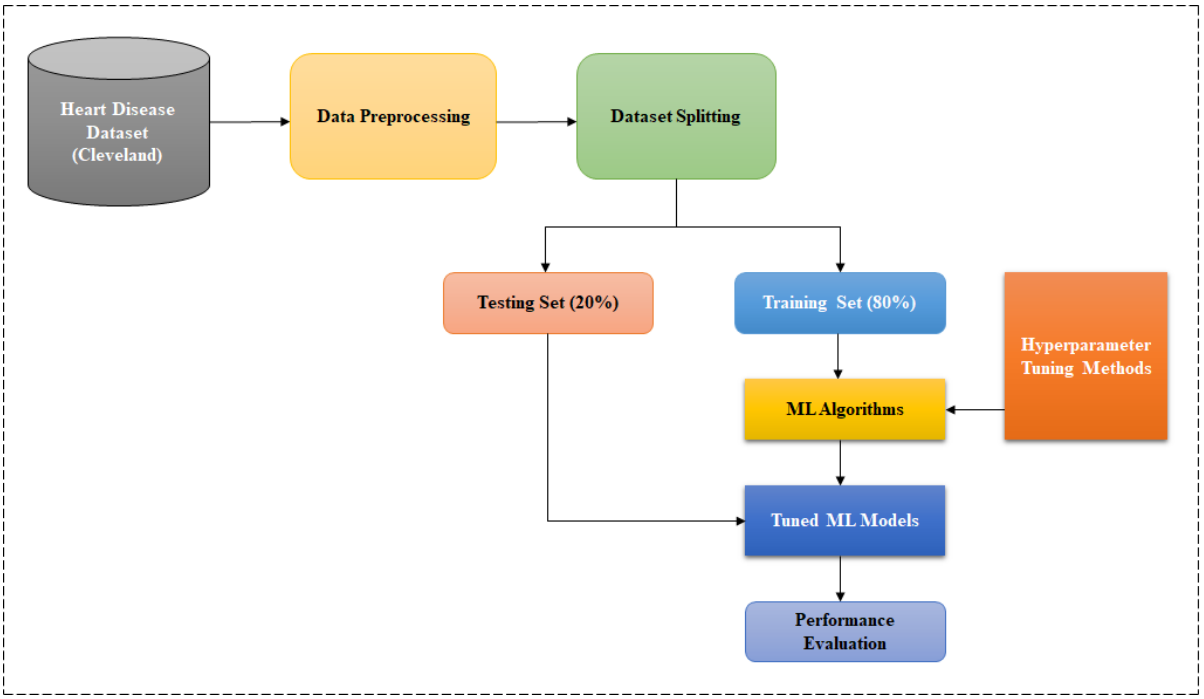

Figure 1. The block diagram of the proposed model

In the data preprocessing stage, the Cleveland Heart Disease dataset [17] is utilized, which consists of 14 key attributes. These features are comprehensively listed in Table 1, which outlines the complete schema of the dataset. The dataset comprises a total of 14 attributes, including the target variable labelled as NUM. This target feature is binary, where a value of 1 indicates the presence of heart disease, and 0 denotes the absence of the condition. The dataset contains 303 patient records, with 164 instances classified as heart disease positive (sick) and 139 as heart disease negative (normal). Among the 14 attributes, 8 are categorical, and 6 are numerical, providing a balanced mix of qualitative and quantitative data for model training and evaluation. This carefully structured dataset forms the basis for building and evaluating predictive models aimed at early and accurate detection of cardiovascular conditions. The consistent use of these 13 features across studies underscores their diagnostic relevance and suitability for machine learning applications in the medical domain.

Table 1. Features of Cleveland Heart Disease Dataset

Sl. No.	Features	Description	Range
1	Age	Age in years	Age in year [29-77]
2	Sex	Patient sex (Male or Female)	0 for female and 1 for male
3	cp	Chest pain type	Chest pain type (1 = typical angina, 2 = atypical angina, 3 = non-anginal pain, 4 = asymptomatic)
4	trestbps	Resting blood pressure	Resting blood sugar in mm Hg on admission to the hospital [94,200]
5	chol	Serum cholesterol	Serum cholesterol in mg/dl [126,564]
6	fbs	Fasting blood sugar	Fasting blood sugar > 120 mg/dl: 1 = true 0 = false
7	Restecg	Resting electrocardiographic results	The results (0 = normal, 1 = having ST-T wave abnormality, 2 = left ventricular hypertrophy
8	thalach	Maximum heart rate achieved	Maximum heart rate achieved [71,202]

9	exang	Exercise induced angina	Exercise induced angina values [0=no 1=yes]
10	oldpeak	ST depression induced by exercise relative to rest	The values are [0,6.2]
11	slope	The slope of the peak exercise ST segment	The slope of the peak exercise segment: (1 = up sloping,2 = flat, 3 = down sloping)
12	ca	Number of major vessels colored by fluoroscopy	Color of fluoroscopy [0-3]
13	thal	Exercise thallium scintigraphy	Heart condition: 3 = normal, 6= fixed defect, 7=reversible defect
14	num	Response: diagnosis of HD	Class (0 = healthy, 1 = have HD)

Let the dataset be represented as:

$$D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n} \quad (1)$$

Where, $x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_{13}^{(i)}] \epsilon R^{13}$ is the feature vector for the i^{th} patient, $y^{(i)} \epsilon \{0,1\}$ is the corresponding label indicating absence (0) or presence (1) of heart disease, and n=303 is the total number of instances.

Prior to model training, the dataset is split into training and testing subsets in an 80:20 ratio, ensuring that the model is evaluated on unseen data. This split is performed using the train_test_split function from the Scikit-learn (sklearn) library, a widely used toolkit in the machine learning domain.

An important step in preprocessing is feature standardization, which ensures that all input features contribute equally to the model's learning process. Since the dataset contains features with varying units and scales, raw data may introduce bias or imbalance during model training. To address this, standardization is applied using the StandardScaler from sklearn. This transformation centers the features by removing the mean and scaling them to unit variance, thereby enabling the learning algorithms to operate more effectively and converge faster. The standardized value $\hat{x}_j$ is computed as:

$$\hat{x}_j = \frac{x_j - \mu_j}{\sigma_j} \quad (2)$$

Where, $\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_j^{(i)}$ is the mean of the j^{th} feature, $\sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_j^{(i)} - \mu_j\right)^2}$ is the standard deviation. This ensures each feature has **zero mean** and **unit variance**, improving the performance of gradient-based algorithms and distance-based classifiers.

Following preprocessing, the standardized training and testing datasets are used to develop classification models. The key innovation in this study lies in the application of hyperparameter tuning techniques to enhance model performance. By integrating these tuning methods with widely used machine learning classifiers, such as Random Forest (RF), Support Vector Machine (SVM), and others, the study aims to develop a robust and highly accurate predictive model for heart disease identification. The impact of hyperparameter tuning is quantitatively evaluated using standard performance metrics, ensuring a comprehensive analysis of each model's diagnostic capability.

3.1 Hyperparameter Tuning Process

Hyperparameter tuning is a critical step in optimizing machine learning models. Unlike model parameters, which are learned during training, **hyperparameters** are set before training begins and guide the learning process—such as the number of trees in a Random Forest, the regularization parameter in logistic regression, or the kernel type in Support Vector Machines. Selecting the right combination of hyperparameters can dramatically affect a model's predictive performance. This process is typically treated as an **optimization problem** over a defined search space of hyperparameter values. The objective is to find the optimal hyperparameter configuration $\lambda^* \epsilon \Lambda$, where Λ denotes the search space, that maximizes a model evaluation metric $\mathcal{M}$.

$$\lambda^* = \underset{\lambda \epsilon \Lambda}{\operatorname{argmax}}\, \mathcal{M}(f(x;\lambda), D_{val}) \quad (3)$$

Where, $f(x;\lambda)$ is the model trained using hyperparameters λ, D_{val} is validation dataset, and $\mathcal{M}$ is chosen performance metric.

3.1.1 Grid Search

Grid Search is a brute-force technique where the hyperparameter space Λ is discretized into a finite grid $\Lambda_g \subset \Lambda$. The model is trained and evaluated for every combination in this grid. Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ be a vector of k hyperparameters. Suppose each λ_i has n_i discrete values. The total number of configurations is:

$$|\Lambda_g| = \prod_{i=1}^{k} n_i \quad (4)$$

For each combination $\lambda_j \in \Lambda_g$, the model is evaluated using **k-fold cross-validation**:

$$M_{cv}(\lambda_j) = \frac{1}{k}\sum_{i=1}^{k} M(f^{(i)}(x;\lambda), D_{val}^{(i)}) \quad (5)$$

Where $f^{(i)}$ is the model trained on the i^{th} fold of the training data. The best configuration is:

$$\lambda^* = arg \max_{\lambda_j \in \Lambda_g} M_{cv}(\lambda_j) \quad (6)$$

Although exhaustive, Grid Search is computationally expensive, especially as the number of hyperparameters increases.

3.1.2 Random Search

Random Search is a more efficient alternative where instead of evaluating all points in Λ_g, a random subset $\Lambda_r \subset \Lambda_g$ is sampled uniformly. Let m be the number of sampled configurations $m \ll |\Lambda_g|$. Then, $\Lambda_r = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, where, $\lambda_i \sim Uniform(\Lambda_g)$. The same cross-validation procedure is used for evaluation:

$$M_{cv}(\lambda_i) = \frac{1}{k}\sum_{j=1}^{k} M(f^{(j)}(x;\lambda_i), D_{val}^{(j)}) \quad (7)$$

And the optimal hyperparameters are selected as:

$$\lambda^* = arg \max_{\lambda_i \in \Lambda_r} M_{cv}(\lambda_i) \quad (8)$$

Random Search often finds a good configuration much faster than Grid Search, especially when only a few hyperparameters are influential.

3.1.3 Halving Grid Search and Halving Randomized Search

These methods fall under the category of **successive halving algorithms**, which progressively allocate more computational resources to promising hyperparameter configurations while discarding underperforming ones. Let $\Lambda_s \subset \Lambda$ be the initial set of s sampled hyperparameter combinations. Each configuration is evaluated using an increasing amount of resources across r rounds.

Halving Strategy:

- Round 1: Evaluate all s configurations using r_1 resources.
- Round 2: Retain the top $\frac{s}{\eta}$ configurations, increase resources to r_2
- ....
- Round t: Retain $\frac{s}{\eta^{t-1}}$ configurations and evaluate with r_t resources.

Where, $\eta > 1$ is the **halving factor,** $r_t = \eta^{t-1} \cdot r_1$ defines resource scaling.

In **Halving Grid Search**, the configurations come from a fixed grid Λ_g. In Halving Random Search, the configurations are randomly sampled from Λ at each round. The best hyperparameter configuration is:

$$\lambda^* = arg \max_{\lambda_i \in \Lambda_s} M_{r_t}(\lambda_i) \quad (9)$$

Where, M_{r_t} is the performance metric evaluated using the largest resource allocation.

3.2 Machine Learning Algorithms

Machine learning (ML) algorithms are the backbone of predictive modelling in data science and artificial intelligence. These algorithms learn patterns from data and make informed decisions or predictions without explicit programming. In supervised learning, a model is trained on a labeled dataset consisting of input features X and corresponding output labels y. The goal is to learn a function $f: X \rightarrow y$ that maps new, unseen inputs to the correct output class with high accuracy. For this study, five key supervised learning algorithms Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT),

Random Forest (RF), and K-Nearest Neighbors (KNN) are employed to classify heart disease presence based on the Cleveland dataset. Each algorithm is optimized using hyperparameter tuning methods to enhance diagnostic performance.

3.2.1 Logistic Regression

Logistic Regression is a statistical learning method used for binary and multiclass classification problems. It models the probability that a given input $x = (x_1, x_2, \dots, x_n)$ belongs to a particular class using a logistic function. In binary classification, the probability is given by:

$$P\left(y = \frac{1}{x}\right) = \sigma(wx + b) = \frac{1}{1+e^{-(w^T x+b)}} \quad (10)$$

Where, w weight vector, b bias term, and $\sigma(z)$ sigmoid activation function.

The model is trained by minimizing the binary cross-entropy loss:

$$\mathcal{L}(w, b) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} log\hat{y}^{(i)} + \left(1 - y^{(i)}\right)\log\left(1 - \hat{y}^{(i)}\right)] \quad (11)$$

3.2.2 Support Vector Machine

SVM is a powerful classification technique that aims to find the optimal hyperplane that separates data points of different classes with the maximum margin. For a binary classification problem, the decision boundary is defined as:

$$f(x) = wx + b = 0 \quad (12)$$

The objective is to maximize the margin $\frac{2}{\|w\|}$ while ensuring correct classification:

$$y^{(i)}(wx^{(i)} + b) \geqq 1 \quad (13)$$

The optimization problem is:

$$\min_{w,b} \frac{1}{2}\|w\|^2, \text{ subject to } y^{(i)}(wx^{(i)} + b) \geqq 1 \quad (14)$$

3.2.3 Decision Tree

A Decision Tree is a tree-structured classifier where internal nodes represent feature tests, branches represent outcomes, and leaf nodes represent final class labels. The tree partitions the data based on the feature that provides the maximum information gain.

Gini Impurity is computed as:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2 \quad (15)$$

Where, p_i proportion of class i instances in dataset D, C number of classes.

The tree splits the dataset D at a feature A to minimize the weighted impurity:

$$Gain(D, A) = Gini(D) - \sum_{v \,\epsilon\, values(A)} \frac{|D_v|}{D} Gini(D_v) \quad (16)$$

The process continues recursively until a stopping criterion is met.

3.2.4 Random Forest

Random Forest is an ensemble method that builds multiple decision trees and merges their outputs to improve predictive performance and control overfitting. It follows the bagging approach bootstrap aggregation where each tree is trained on a random subset of the data and features. Prediction is done by majority voting:

$$\hat{y}_{final} = mode(\hat{y}_1, \hat{y}_2, \dots., \hat{y}_N) \quad (17)$$

Where, $\hat{y}_i$ is the prediction from the i^{th} tree and N is the total number of trees. Random Forest reduces variance and improves generalization by aggregating diverse tree models.

3.2.5 K-Nearest Neighbors

KNN is a non-parametric, instance-based learning algorithm used for classification and regression. It classifies a new data point based on a majority vote of the k nearest data points in the feature space using a distance metric.

$$d(x, x') = \sqrt{\sum_{i=1}^{n}(x_i - x_i')^2} \quad (18)$$

The class label is determined by:

$$\hat{y} = arg\max_{c \,\epsilon\, C} \sum_{i \,\epsilon\, N_k(x)} 1(y^{(i)} = c) \quad (19)$$

Where, $N_k(x)$ set of k nearest neighbors, $1(.)$ indicator function. The choice of k affects model bias-variance. A small k

can lead to overfitting, while a large k increases bias.

Table 2 below presents and explains the various hyperparameters utilized in the implementation of machine learning algorithms. These hyperparameters are fine-tuned using hyperparameter optimization techniques to achieve the highest classification accuracy for heart disease detection using the Cleveland Heart Disease dataset. The table also highlights the optimal estimated values for each hyperparameter.

Table 2. Algorithms and Tuned Hyperparameters

Algorithm	**Key Hyperparameters**	**Tuned Using**
Logistic Regression	Regularization strength (C), Solver	Grid Search, Random Search
SVM	Kernel type, C, Gamma	Grid Search, Random Search
Decision Tree	Max depth, Min samples split	Grid Search
Random Forest	Number of trees, Max features	Halving Grid/Random Search
KNN	Number of neighbors (k), Distance metric	Grid Search, Random Search

4. RESULTS AND DISCUSSIONS

The experimental results obtained from the implementation phase are comprehensively summarized in Table 3. These results demonstrate the effectiveness of various machine learning algorithms when integrated with different hyperparameter tuning strategies. The models were evaluated based on their classification accuracy in identifying heart disease using the Cleveland Heart Disease dataset. To ensure consistency and reliability, each algorithm underwent tuning through four optimization techniques: Grid Search, Random Search, Halving Grid Search, and Halving Randomized Search. These tuning methods helped in identifying the optimal set of hyperparameters for each model, contributing to variations in their final predictive performance.

Figure 2 visually illustrates the comparative performance of each machine learning model under the different hyperparameter tuning strategies. It is evident from both the table and figure that the Random Forest classifier, when tuned using Random Search, achieved the highest accuracy of 92.45%, making it the most effective model among those evaluated. On the other hand, the K-Nearest Neighbors (KNN) algorithm yielded the lowest accuracy of 64%, highlighting its sensitivity to parameter settings and possibly the influence of data distribution in the feature space. This comparison underscores the critical importance of selecting appropriate algorithms and tuning methods to maximize predictive performance in medical diagnosis applications.

Table 3. Accuracy comparison of ML algorithms - with and without hyperparameters tuning

Classifiers	**Without Hyper-Parameter Tuning**	**Hyper-Parameter Tuning**			
		Grid Search	**Random search**	**Halving Grid**	**Halving Random**
	Accuracy	**Accuracy**	**Accuracy**	**Accuracy**	**Accuracy**
RF	**83%**	**88%**	**92.45%**	**81%**	**83%**
SVM	67%	87%	87%	81%	75%
DT	78%	82%	79%	75%	78%
Logistic Regression	75%	82%	85%	74%	75%
KNN	65%	67%	64%	76%	74%

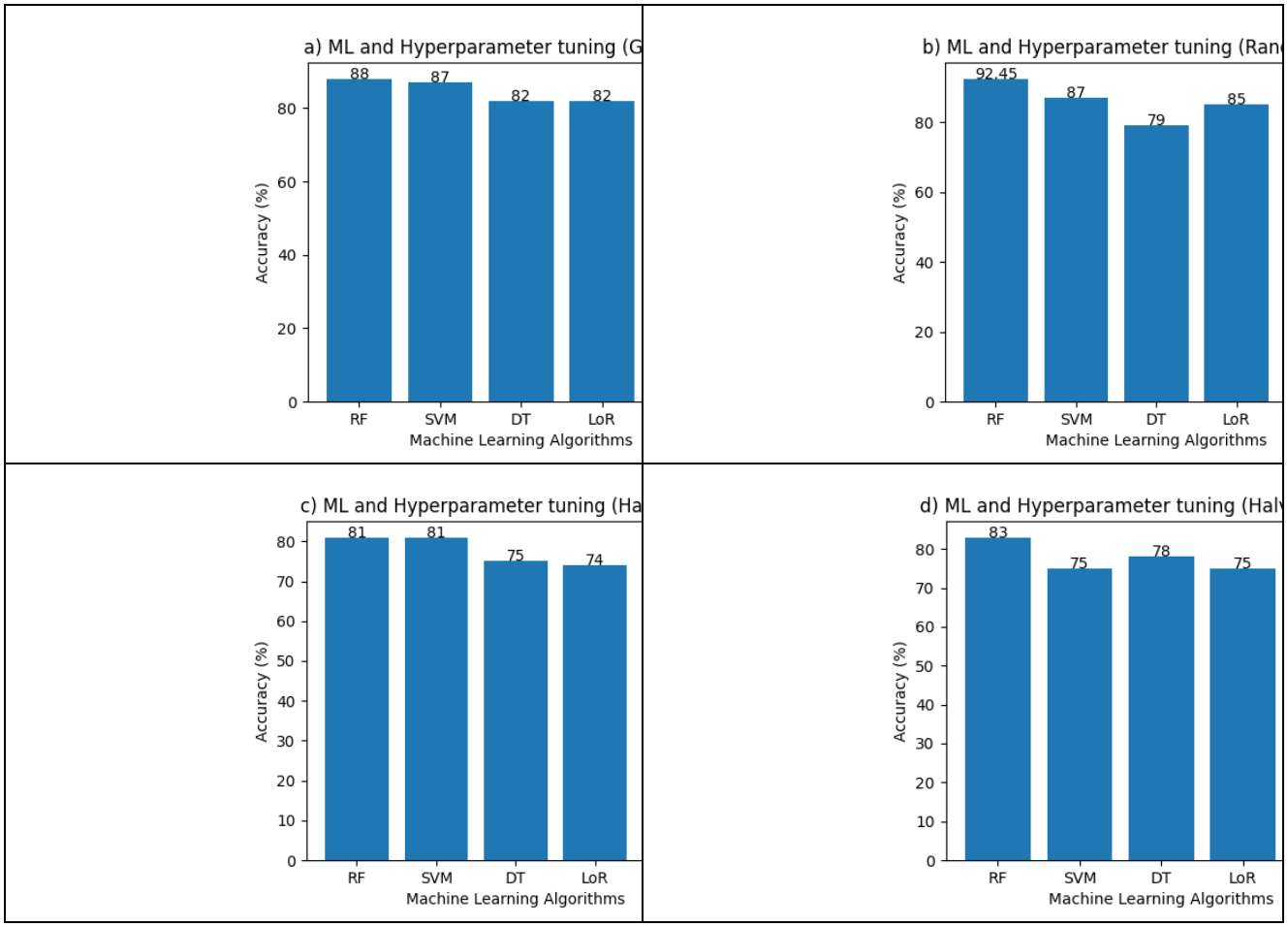

Figure 2. Performance of ML models with various hyperparameter tuning methods

5. CONCLUSION

The proposed framework significantly improves the predictive performance of machine learning algorithms by incorporating advanced hyperparameter tuning techniques. For experimental validation, the Cleveland Heart Disease dataset from the UCI Machine Learning Repository was employed. A range of machine learning classifiers namely Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), K-Nearest Neighbors (KNN), and Artificial Neural Network (ANN) were utilized to detect the presence of heart disease. Each of these models was optimized using various hyperparameter tuning methods, including Grid Search, Random Search, Halving Grid Search, and Halving Randomized Search.

Comprehensive experimental analysis reveals that the combination of Random Search (RS) tuning with the Random Forest (RF) classifier delivers the most accurate and robust results. Among all the tested configurations, this hybrid approach RS+RF achieved the highest classification accuracy of 92.45%, outperforming all other model and tuning method combinations. These findings underscore the effectiveness of strategic hyperparameter optimization in enhancing the diagnostic capability of machine learning models in medical applications

REFERENCES

[1] World Health Organization. (2021). Noncommunicable diseases. Retrieved from https://www.who.int/news-room/fact-sheets/detail/noncommunicable-diseases

[2] Giri, R., & Shukla, A. (2020). Heart disease prediction using machine learning techniques: A survey. International Journal of Computer Applications, 175(21), 7–11.

[3] Haq, A. U., et al. (2018). Hybrid machine learning model for heart disease prediction using data mining techniques. International Journal of Engineering & Technology, 7(2.8), 290-293.

[4] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.

[5] Johnson, A. E. W., et al. (2016). Machine learning and decision support in critical care. Proceedings of the IEEE, 104(2), 444–466.

[6] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In Advances in Neural Information Processing Systems, 1(1), 2951–2959.

[7] Li, L., Jamieson, K. G., Rostamizadeh, A., Gonina, E., Hardt, M., Recht, B., & Talwalkar, A. (2020). A system for massively parallel hyperparameter tuning. Communications of the ACM, 63(5), 90-99.

[8] Jinny, S., Vinila, S., & Mate, Y. V. (2021). Early prediction model for coronary heart disease using genetic algorithms, hyper-parameter optimization and machine learning techniques. Health and Technology, 11(1), 63–73.

[9] Asif, M., Islam, M. T., Hossain, M. S., & Hossain, M. A. (2021). Performance evaluation and comparative analysis of different machine learning algorithms in predicting cardiovascular disease. Engineering Letters, 29(2).

[10] Hashi, E. K., & Zaman, M. S. U. (2020). Developing a hyperparameter tuning based machine learning approach of heart disease prediction. Journal of Applied Science & Process Engineering, 7(2), 631–647.

[11] Hashi, E. K., & Zaman, M. S. U. (2020). Developing a hyperparameter tuning based machine learning approach of heart disease prediction. Journal of Applied Science & Process Engineering, 7(2), 631–647.

[12] Firdaus, F. F., Nugroho, H. A., & Soesanti, I. (2021). Deep neural network with hyperparameter tuning for detection of heart disease. In 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob) (pp. 1–6). IEEE.

[13] Sonth, M. V., Padma, M. C., & Reddy, K. R. (2020). Optimization of random forest algorithm with ensemble and hyper parameter tuning techniques for multiple heart diseases. Solid State Technology, 63(5), 3961–3972.

[14] Valarmathi, R., & Sheela, T. (2021). Heart disease prediction using hyper parameter optimization (HPO) tuning. Biomedical Signal Processing and Control, 70, 103033.

[15] El-Shafiey, M. G., Mohamed, A. W., Ahmed, A. Y., & El-Metwally, S. E. D. (2022). A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest. Multimedia Tools and Applications, 81(13), 18155–18179.

[16] Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Proceedings of the 30th International Conference on Machine Learning (pp. 1–9).

[17] https://archive.ics.uci.edu/ml/datasets/heart+disease.