

Refining heart disease prediction accuracy using hybrid machine learning techniques with novel metaheuristic algorithms

Haifeng Zhang^a, Rui Mu^{b,*}

^a The first people's Hospital of Baiyin, Baiyin, Gansu 730900, China

^b The second people's Hospital of Baiyin, Baiyin, Gansu 730900, China

ARTICLE INFO

Keywords:

Heart failure
Machine learning
Meta-heuristic algorithms
Medical technology

ABSTRACT

Early diagnosis of heart disease is crucial, as it's one of the leading causes of death globally. Machine learning algorithms can be a powerful tool in achieving this goal. Therefore, this article aims to increase the accuracy of predicting heart disease using machine learning algorithms. Five classification models are explored: eXtreme Gradient Boosting (XGBC), Random Forest Classifier (RFC), Decision Tree Classifier (DTC), K-Nearest Neighbors Classifier (KNNC), and Logistic Regression Classifier (LRC). Additionally, four optimizers are evaluated: Slime mold Optimization Algorithm, Forest Optimization Algorithm, Pathfinder algorithm, and Giant Armadillo Optimization. To ensure robust model selection, a feature selection technique utilizing k-fold cross-validation is employed. This method identifies the most relevant features from the data, potentially improving model performance. The top three performing models are then coupled with the optimization algorithms to potentially enhance their generalizability and accuracy in predicting heart failure. In the final stage, the shortlisted models (XGBC, RFC, and DTC) were assessed using performance metrics like accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC). This rigorous evaluation identified the XGGA hybrid model as the top performer, demonstrating its effectiveness in predicting heart failure. XGGA achieved impressive metrics, with an accuracy, precision, recall, and F1-score of 0.972 in the training phase, underscoring its robustness. Notably, the model's predictions deviated by less than 5.5 % for patients classified as alive and by less than 1.2 % for those classified as deceased compared to the actual outcomes, reflecting minimal error and high predictive reliability. In contrast, the DTC base model was the least effective, with an accuracy of 0.840 and a precision of 0.847. Overall, the optimization using the GAO algorithm significantly enhanced the performance of the models, highlighting the benefits of this approach.

1. Introduction

1.1. Basic overview

Presently, early detection of heart disease plays a crucial role in averting heart failure, particularly among older individuals [1,2]. When the heart muscles enlarge and weaken, they struggle to pump blood efficiently, ultimately leading to heart failure [3–5]. Consequently, the heart's ventricles, designed to flex between beats, fail to adequately fill [6]. Over time, this inability of the heart to meet the body's demand for blood results in breathing difficulties, a concerning symptom [7]. This incapacity of the heart to sufficiently supply blood reflects its diminished power [8], thereby restricting the delivery of oxygenated blood to meet the body's requirements [9]. While a diseased heart remains

functional, its efficiency weakens in comparison to a healthy one. Notably, several fundamental complications contribute to heart failure, including coronary heart disease, diabetes, and high blood pressure, as well as other conditions like HIV, thyroid disorders, and excessive vitamin E levels [10]. Consequently, research into diagnosing heart failure holds significant importance [11,12].

Generally, medical diagnosis is crucial, even though a complex endeavor imposing accuracy and effectiveness [13]. Nonetheless, there is an urgent requirement for substantial progress in both diagnosing and treating heart diseases to attain optimal levels of accuracy. Accessing extensive medical data is also imperative, given the remarkable tools available for data analysis and knowledge extraction [14]. Presently, the utilization of computer-aided methods has greatly streamlined the diagnostic process, proving beneficial in terms of accuracy and

* Corresponding author.

E-mail address: muruiok2022@163.com (R. Mu).

reliability [15,16]. Consequently, algorithms leveraging Artificial Intelligence (AI) and Machine Learning (ML) are instrumental in diagnosis because of their computational proficiency [17,18]. These methods analyze medical images to identify latent human conditions, particularly those unapparent to experts. The capabilities of AI and ML algorithms enable the early detection of diseases such as Alzheimer's or cancer before clinical diagnosis [19]. In identifying patients at risk of heart failure, physicians typically evaluate for excess lung fluid using X-ray images [20]. The presence of lung-related edema, characterized by water accumulation in the lungs, impairs blood oxygenation and confronts significant risks [21]. However, these examinations often rely on subtle characteristics, bringing out potential misdiagnosis and inappropriate treatment plans [22,23].

In today's modern era, the digitalization of images has become familiar. To improve the results, adjustments to these images seem necessary [24,25]. These modifications typically fulfill three primary objectives: processing, analyzing, and perceiving the image [26]. Thus, computer systems for image processing have been developed to execute these tasks swiftly and accurately. Within these systems, four primary processes are employed: preprocessing, image quality improvement, conversion of images, and classification and analysis of images [27]. These methods comprise the formulation of rules to imitate human vision elements by the computer, employing mathematical approaches [28]. This aspect of image analysis fulfills specific objectives [27]. Computer Vision, a scientific field dedicated to analyzing images, is applied in diverse fields, surpassing medicine, engineering, molecular imaging, astronautics, security, etc. Image processing is notably prevalent in medical contexts [29]. For instance, various diseases can be diagnosed at a low cost in the fastest possible time by utilizing image processing. Consequently, authors can employ image processing systems as aids for medical practitioners [30]. These systems can extract pertinent information from images, facilitating straightforward and rapid disease analysis by doctors. A primary advantage of medical image processing lies in its ability to enable thorough yet non-invasive examinations of internal anatomy. This technology permits the creation of 3D anatomical models, leading to enhanced treatment outcomes for patients, the advancement of medical devices and drug delivery systems, and more informed diagnoses, thus playing a pivotal role in recent medical advancements [31]. Continuous advancements in medical image processing, coupled with sophisticated software tools, have enabled precise digital reconstruction of anatomical structures at various scales and with diverse characteristics, including bones and soft tissues. This encompasses measurements, statistical analysis, and the generation of simulated models that closely resemble real anatomical geometry. Such advancements provide valuable insights into the interaction between patient anatomy and medical devices [32].

1.2. Literature survey

Many studies focus on the issue of diagnosing heart failure by the ML method [33–40]. A supervised machine-learning algorithm was employed by Princy et al. [41] to categorize the dataset and predict cardiac illness. The outcomes highlighted the superiority of Decision Tree classification over Naive Bayes, Logistic Regression, Random Forest, SVM, and KNN-based methods with an accuracy of 73 %. In an examination, Shorewala [42] utilized a risk factor method to anticipate coronary heart disease (CHD). Their suggested method depicted superior performance in comparison with K-Nearest Neighbors, Binary Logistic Classification, and Naive Baye, particularly in essential metrics, including recall and ROC curves. Thereafter, the performance of the tested models was verified by employing data-analytic methods and K-Folds cross-validation. Tasnim and Habiba [43] attempted to investigate the probability of CHD by data mining classification methods, namely NB, SVM, kNN, DT, Neural Network, LR, and Random Forest. The achieved outcomes demonstrated that Random Forest, with a precision of 92.85 %, performs the most desired in heart disease classification. An

innovative ML method with an accuracy of 94.2 % was suggested by Bharti et al. [44] for investigating heart disease, and a confusion matrix verified the achieved outcomes matrix. In addition, the classification of cardiovascular disease was presented by random forest, and the results indicated the correlation between diabetes and heart failure [45].

Rizinde et al. [46] compared six machine learning models for predicting heart failure readmission within 20 days post-discharge. Their results indicated that the random forest (RF) outperforms all others with a 94 % AUC, while decision trees (DT) performed poorly with an AUC of 57 %. Boukhatem et al. [47] utilized several classification methods, including Multi-layer Perceptron (MLP), SVM, RF, and NB, for predicting Cardiovascular illness. Regarding the data preprocessing and feature selection, SVM models with an accuracy of 91.67 % represented notable performance. Bhatt et al. [48] examined the significance of accurate cardiovascular disease (CVD) diagnosis, suggesting an ML approach, such as an RF for prediction. Their model, incorporating k-mode clustering with Huang initialization, improved classification accuracy significantly. Utilizing various ML models, they achieved notable accuracies. Particularly, the multilayer perceptron with cross-validation outperformed others with 87.28 % accuracy.

Khan et al. [49] employed ML algorithms for precise CVD prediction and decision-making. They sampled heart disease patients from Pakistani hospitals and tested various ML methods, including DT, RF, LR, NB, and SVM. Ultimately, RF demonstrated the maximum accuracy, sensitivity, and ROC curve at 85.01 %, 92.11 %, and 87.73 %, respectively, with the minimum specificity and misclassification errors at 43.48 % and 8.70 % for CVD. Kumar et al. [50] examined the effect of changing lifestyles on physical activity and increased heart-related diseases. They suggested an Edge-assisted Cloud-IoT framework with a Random Forest and Logistic Regression Grid (RF-LRG) approach for real-time heart disease analysis. This framework outperformed traditional algorithms improved accuracy and reduced delay and energy consumption compared to cloud or edge computing alone.

Chandrasekhar et al. [51] utilized machine learning to improve heart disease prediction accuracy. They examined six algorithms on Cleveland and IEEE Dataport datasets, with logistic regression and AdaBoost excelling. Through a soft voting ensemble classifier, remarkable accuracy improvements were achieved, surpassing individual models. Their methodology, including GridSearchCV and five-fold cross-validation, proved innovative and outperformed existing prediction studies. Saranya and Pravin [52] proposed RF-FSFC, a method for improving heart disease prediction employing the Cleveland dataset. By integrating feature sensitivity and correlation analysis into RF, they achieved 81.16 % accuracy with feature omission and 86.141 % without. Their approach performed more desired in comparison with other models, offering superior sensitivity, specificity, PPV, and NPV scores, highlighting significant accuracy improvement. Das et al. [53] conducted a study on strokes, a leading cause of death globally, often stemming from blockages in heart and brain pathways. They highlighted the importance of early diagnosis and explored various risk factors. Utilizing ML methods, they aimed to predict heart attacks, testing nine algorithms. The RF method emerged as the top performer, achieving 98.4 % accuracy.

Ansari et al. [54] suggested a method for predicting heart disease using various machine-learning algorithms. They found that the k-nearest neighbor and random forest approaches outperformed others, achieving a remarkable 99.04 % accuracy rate. Additionally, they employed six feature selection algorithms and evaluated performance using MCC parameters. Ashish et al. [55] assessed the effectiveness of SVM and XGBoost methods in identifying ischemic heart disease, yielding noteworthy outcomes. Despite this interest, the ongoing competition among machine learning models in predicting heart diseases remains unresolved. Consequently, additional research is necessary to achieve heightened accuracy.

Table 1

The feature data is based on inputs and outputs.

	Features	ID
Inputs	Age	1
	Ca	2
	Chol	3
	Cp	4
	Exang	5
	Fbs	6
	Oldpeak	7
	Restecg	8
	Sex	9
	Slope	10
Output	Thal	11
	Thalach	12
	Trestbps	13
	Heart Disease	14

1.3. Main contributions and novelties

This study pushes the boundaries of heart failure prediction in a few key ways. Firstly, it integrates feature selection and optimization algorithms. Most studies utilize these techniques independently, but here they'll work in tandem. Selecting the most impactful features through k-fold cross-validation and then fine-tuning them with optimization algorithms has the potential to create models with superior accuracy and generalizability. Secondly, the study delves into a less-explored area by using a diverse set of optimization algorithms. While established methods exist, this research investigates the effectiveness of four less common algorithms (Slime mold, Forest, Pathfinder, and Giant Armadillo) specifically for heart failure prediction. Evaluating these alongside traditional approaches can provide valuable insights into their suitability for this domain. Thirdly, rather than relying solely on a single top performer, the study proposes developing hybrid models. The top three models identified through feature selection and evaluation will be coupled with optimization algorithms. This approach capitalizes on the strengths of various machine learning techniques and optimization algorithms, potentially leading to an enhanced resilience and accurate forecasting model for heart failure. Finally, the study has the potential to translate into real-world clinical applications. If a highly accurate model for predicting heart failure emerges, it could significantly improve clinical practices. Early and precise prediction empowers physicians to intervene sooner, potentially leading to better patient outcomes.

2. Description of case study

This research seeks to conduct a comprehensive analysis of heart disease. The models used in this research are thirteen input parameters and one output variable, which are explained below. Also, in the next step, the features selected to continue the work will be described.

2.1. Selected data

The dataset used in this study contains crucial patient information related to heart disease, extracted from Matt Hartman's research (<https://www.kaggle.com/datasets/hartman/heart-disease-uci>). It consists of 13 key input parameters that play a significant role in understanding heart health: age in years (age), sex, chest pain type (cp), resting blood pressure (trestbps), serum cholesterol (chol), fasting blood sugar (fbs), resting electrocardiographical results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise relative to rest, which indicates the stress on the heart during physical activity (oldpeak), the slope of the peak exercise ST segment (slope), the number of major vessels (ca), and thallium stress result (thal). Additionally, the dataset includes one output parameter that determines the presence or absence of heart disease. These parameters are considered as variables that can significantly influence the

mortality rate due to heart disease. By analyzing the relationships and interactions among these variables, researchers can gain valuable insights for future studies and medical interventions. [Table 1](#) provides an in-depth overview of these features and parameters, while [Fig. 1](#) visually represents the surface plot for the correlation of the inputs and output variables. This visual representation helps in understanding their predictive utility in identifying mortality risks associated with heart disease.

2.2. Feature selection

Feature selection involves meticulously selecting a subset of pertinent features from a dataset to enhance the effectiveness of machine learning models. Its main goals are to mitigate overfitting, lower computational expenses, and enhance interpretability. By opting for the most informative features, feature selection simplifies model training, reduces complexity, and facilitates a more profound comprehension of the inherent data patterns. Essentially, it acts as a guiding principle, directing machine learning efforts toward more efficient and interpretable models. In this study, according to the obtained results and the importance of their feature position, six input parameters of age, sex, cp, trestbps, chol, and fbs have been selected as main and important features. Various methods can be employed for variable selection, including:

- Filter Methods:** These approaches evaluate the significance of features by considering statistical properties of the data, such as their correlation with the target variable, mutual information, or the outcomes of significance tests like F-tests.
- Wrapper Methods:** Wrapper methods evaluate machine learning models' performance with different feature subsets, often yielding better results than filter methods despite being more computationally demanding.
- Embedded Methods:** These methods integrate feature selection into the model training process.
- Dimensionality Reduction Techniques:** Dimensionality reduction techniques are approaches utilized to decrease the number of input variables or features within a dataset. They are commonly used in the context of attribute selection to enhance the performance of machine learning models. These techniques are designed to combat the curse of dimensionality, which encompasses the difficulties encountered when dealing with high-dimensional data, including heightened computational complexity and the potential for overfitting. Techniques such as principal component analysis (PCA) and t-SNE random embedding (t-SNE) reduce the dimensionality of the feature space by transforming the original features into a lower-dimensional representation. Meanwhile, PCA technique can be one of the first choices in this field. PCA is a dimensionality reduction technique used in data analysis and machine learning to transform datasets into a new set of uncorrelated variables called principal components. These components are ordered by the amount of variance they explain, with the first component explaining the most variance. The features of this technique are summarized below.

- Reduces the number of dimensions in a dataset while preserving its variance.
- Simplifies complex datasets, aiding in visualization and analysis.
- Helps in reducing noise and redundancy, which can enhance the effectiveness of models for machine learning.
- Can be used for data visualization, outlier detection, and pre-processing.
- Is a linear transformation technique and may not be suitable for datasets with non-linear relationships.

[Fig. 2](#) includes the results obtained for the selected features, among the thirteen desired parameters, finally, six features were selected

according to the importance of the position they have obtained. In the continuation of this section, k-means clustering results are shown. As we know, K-means is a popular clustering algorithm used in machine learning to partition a data set into K distinct and non-overlapping clusters. This algorithm has different stages such as Initialization, Assignment, Update, and Iteration, which ultimately aims to minimize intra-cluster variance, which is the sum of the squared distances between each data point and its assigned center. In the figure shown, the algorithm's response for $k = 2$ is more optimal than the other modes. Finally, this figure investigated the forms associated with PCA with arrows for closely related features. In the analysis of PCA results is better that consider these items.

- Direction of Arrows

- Length of Arrows
- Similarity in Direction and Length
- Identifying Feature Groups
- Interpreting Feature Relationships

Therefore, hence in the corresponding figures of “PCA Arrows for Closely Related Features” and due to the PCA features, features 1,3,8,10,12, and 13 are selected. Features with an importance score higher than 0.08 were selected as input variables. These features includes are age, cp, thalach, oldpeak, ca, and thal.

3. Methodology

To examine the probability of death and survival in people suffering

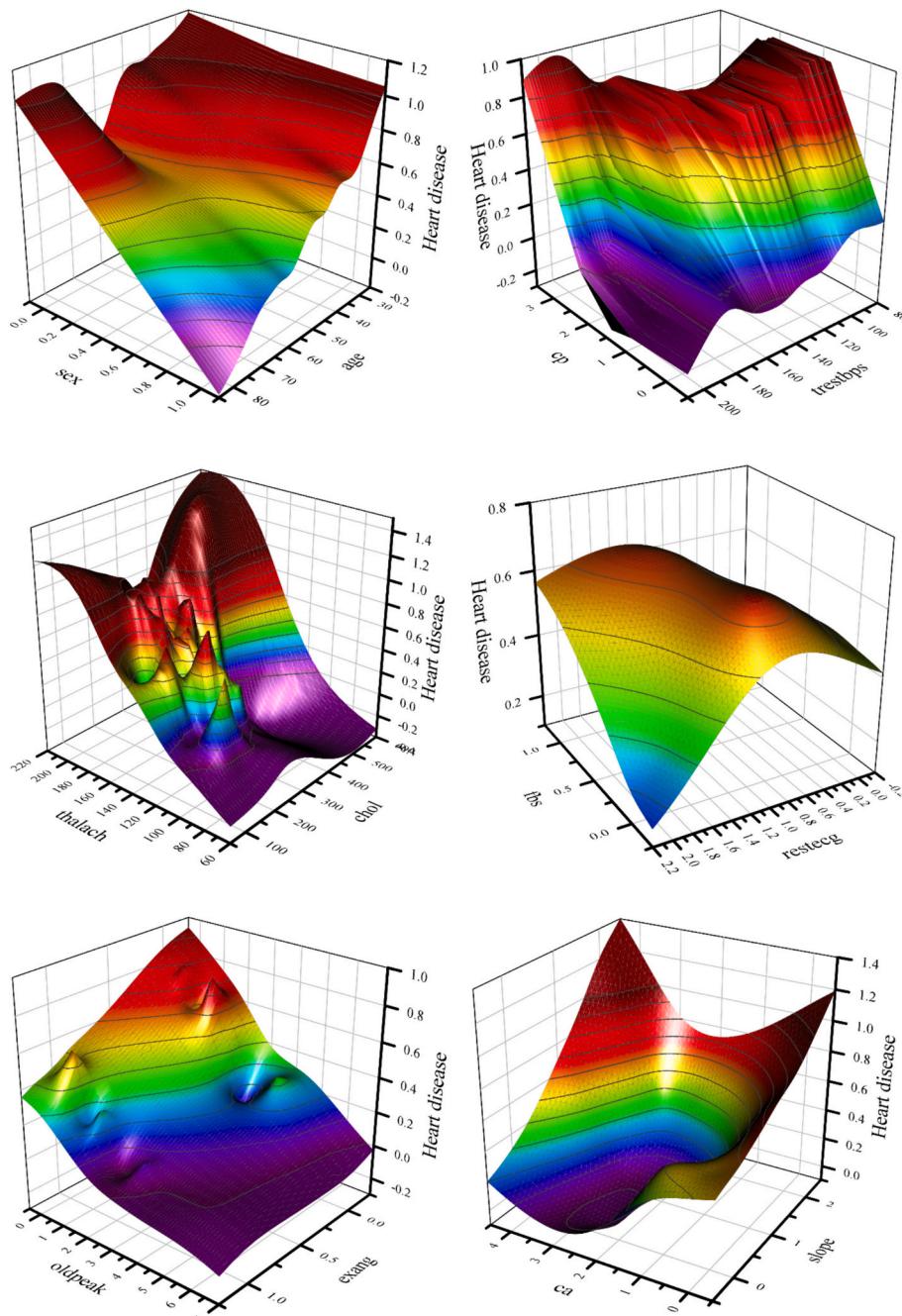


Fig. 1. Surface plot for the correlation of the inputs and output variables

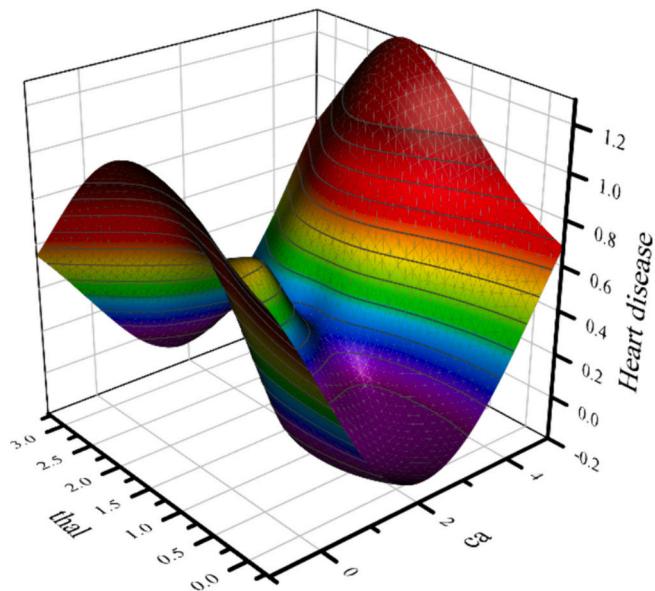


Fig. 1. (continued).

from heart failure, five models according to ML, namely Random Forest Classifier (RFC) [56], Logistic Regression Classification (LRC) [57], K-Nearest neighbor Classifier (KNNC) [58], Extreme Gradient Boosting Classifier (XGBC) [56], and Decision Tree Classifier (DTC) [59] are selected. A random forest operates as a meta-estimator by leveraging multiple decision tree classifiers applied to distinct sections within the data set. This approach enhances predictive accuracy by amalgamating the outcomes of these individual classifiers while simultaneously mitigating overfitting. Notably versatile, the RFC is adept at addressing a wide range of tasks, including both regression and classification, making it a valuable tool in machine learning applications. LRC in machine learning pertains to evaluating the effectiveness of cancer treatments, particularly in containing the disease within its original area without spreading to nearby regions. It involves analyzing medical imaging data, like MRI or CT scans, to assess treatment outcomes. Machine learning algorithms can aid in this analysis by identifying changes in tumor characteristics, helping guide medical decisions, and predicting treatment success. Meanwhile, KNNC is a straightforward algorithm used for classification and regression tasks in machine learning. During training, it stores all training instances with their labels. When predicting a new instance, it finds the k nearest training instances and uses them to predict the label or value. It's non-parametric and lazy, making no assumptions about data distribution and deferring computation to prediction time.

The choice of k affects performance, and it may not perform well in high-dimensional spaces due to computational complexity and the “matter of dimensionality.” Despite this, k-NN remains popular for its simplicity and effectiveness in various tasks. XGBoost classification is a powerful and popular machine learning algorithm known for its exceptional performance in various classification tasks. It belongs to the ensemble learning methods, specifically the gradient boosting framework, which sequentially adds weak learners to improve model accuracy. XGBoost optimizes the gradient boosting algorithm by utilizing a scalable and efficient approach. It employs a technique called gradient boosting, which iteratively builds a collection of decision trees, with each subsequent tree learning to correct the errors of its predecessors. XGBoost introduces several enhancements over traditional gradient boosting methods, including regularization techniques to prevent overfitting, parallel processing capabilities for faster training, and support for handling missing data. Also, the XGBoost classification stands out for its high predictive accuracy, robustness, and efficiency, making it

a go-to choice for many data scientists and machine learning practitioners. A DTC is a machine learning algorithm used for classification and regression tasks. It constructs a tree-like structure where each internal node represents a feature and each leaf node represents a class label or predicted value. The algorithm selects the best feature to split the dataset at each node, based on criteria like purity or error reduction. Decision trees are easy to interpret and don't require extensive preprocessing, but they can suffer from overfitting and instability. Despite these limitations, decision tree classifiers are widely used due to their simplicity and the capacity to manage numerical and category data.

It's crucial to emphasize that the mentioned models necessitate optimization through a variety of algorithms to enhance their accuracy and efficiency. Among the optimizers utilized are the Giant Armadillo Optimization (GAO) [60], Slime Mold Algorithm (SMA) [61], Pathfinder Algorithm (PFA) [62], and Forest Optimization Algorithm (FOA) [63]. These optimization techniques are employed to fine-tune the parameters and configurations of the models, ensuring they perform optimally across diverse datasets and tasks. By integrating these advanced optimization methods, the models can effectively adapt to the complexities of real-world data and deliver superior performance in their respective domains.

The selection of the five machine learning algorithms eXtreme Gradient Boosting (XGBC), Decision Tree Classifier (DTC), Random Forest Classifier (RFC), Logistic Regression Classifier (LRC) and K-Nearest Neighbors Classifier (KNNC) was based on their proven effectiveness and relevance in heart disease prediction. These algorithms were chosen for their distinct advantages, each contributing to a comprehensive evaluation of predictive performance. XGBC and RFC were selected due to their powerful ensemble learning capabilities. Ensemble methods like these combine the predictions of multiple base models, leading to improved accuracy and robustness. XGBC is recognized for its efficiency and strong performance across complex datasets, making it a top choice in modern machine learning applications. Similarly, RFC is favored for its ability to handle large datasets with high-dimensional features and its resilience to overfitting, providing reliable predictions even in challenging scenarios. DTC was chosen for its simplicity and interpretability. Decision trees offer a straightforward approach to classification by segmenting data into subsets based on feature values, creating an easily understandable model.

Despite its simplicity, DTC often serves as a strong baseline model, making it valuable for comparison against more complex algorithms.

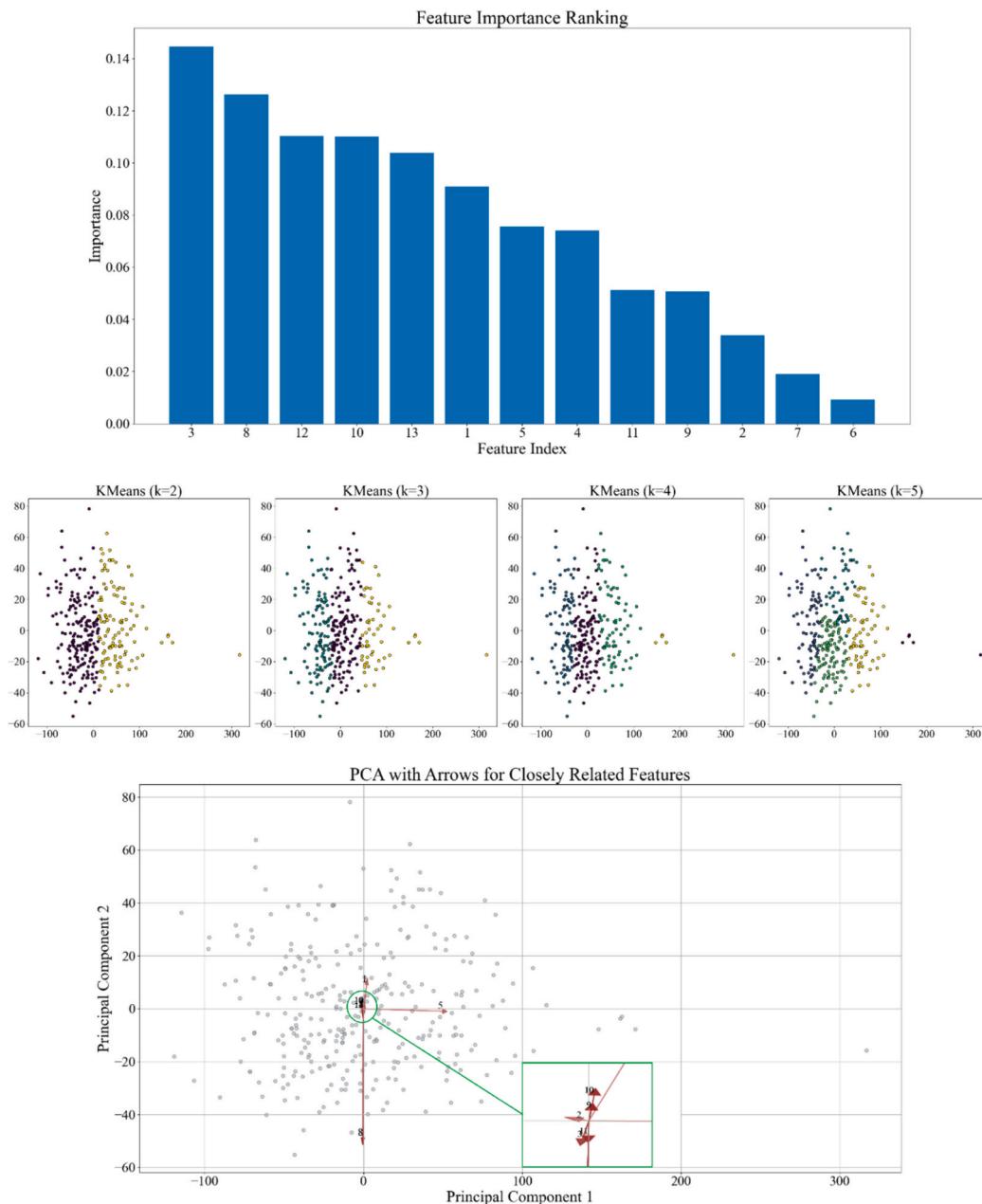


Fig. 2. PCA features selection results of the input parameters

KNNC was included for its non-parametric nature, which is well-suited for scenarios where decision boundaries between classes are not linear or easily separable. The k-nearest neighbors algorithm considers the proximity of data points, making it effective in capturing local patterns in the data that other algorithms might overlook. LRC was selected due to its foundational role in binary classification tasks. Logistic regression is one of the most widely used algorithms due to its simplicity, interpretability, and effectiveness in predicting binary outcomes. Its incorporation offers a starting point for assessing the efficiency improvements brought about by higher-level models. These five algorithms were chosen not only for their individual strengths but also to represent a diverse set of modeling approaches, including ensemble methods, tree-based models, distance-based classifiers, and linear models. This diversity ensures a comprehensive assessment of different machine learning techniques and their applicability to heart disease prediction, enabling a more robust comparison and analysis of model performance.

3.1. Random Forest Classifier (RFC)

A random forest (RF) model is formed by amalgamating a collection of tree predictors. The construction of each tree adheres to the process outlined in:

1. During the bootstrap phase, a random subset of the training dataset is chosen as a local training set for tree growth [64,65]. The remaining samples in the training dataset make up an out-of-bag (OOB) set, which is used to estimate the goodness-of-fit of the random forest (RF) [66].
2. Throughout the growth phase, the tree expands by dividing employing a variable value picked at random from a selection of variables, the local set of training variables at each node. The best split procedure uses a classification and regression tree (CART) approach [67].

3. Each tree is grown to its extreme capacity, and no trimming is applied.

The growth and bootstrap stages involve introducing chance elements. Presumably, these components possess independence among follow and trees and possess a consistent dispersion. Consequently, every tree may be regarded as a separate sample drawn from the whole collection of tree predictors for a particular training dataset. An instance travels through every tree in a forest during the prediction process until it reaches a terminal node, which at that point gives it a class [68]. Each tree's forecast is aggregated through a voting procedure, and the forest finally designates a class according to which tree received the most votes [68]. In the event of draws, a class is randomly selected for resolution [69]. In the upcoming section, this research will introduce a feature contribution procedure, which necessitates establishing a probabilistic interpretation for the forest prediction process. Let $C = \{C_1, C_2, \dots, C_K\}$ indicate the set of classes, and ΔK represent the set.

$$\Delta_k = \left\{ (P_1, \dots, P_K) : \sum_{k=1}^K P_k = 1 \text{ and } P_k \geq 0 \right\} \quad (1)$$

An element in ΔK can be understood as a probability distribution across C . Let e_k be an element of ΔK , with a value of 1 at position k , suggesting A distribution of probabilities with the class C_k as its focus. Should a tree t forecast that an instance i is a member of the C_k , it is represented as $Y^i; t = e_k$. This establishes a mapping from tree predictions to the set ΔK of probability measures on C .

$$\widehat{Y}_k = \frac{1}{T} \sum_{t=1}^T \widehat{Y}_{it} \quad (2)$$

In Eq. (2), T represent the total number of trees in the forest. Therefore, Y^i is an element of ΔK , and for instance i , the RF prediction corresponds to class C_k , where Y^i has the greatest $k - th$ coordinate.

3.2. Logistic regression (LRC)

LR that is Logistic Regression is a regression method developed to forecast a binary dependent variable. The LR equation is formulated using the maximum-likelihood ratio to evaluate the statistical significance of the variables, as described in [70,71]. LR is valuable when using a set of predictor variables to forecast whether a specific trait or result will exist or not. While LR and linear regression are similar, LR excels in situations in cases where the binary dependent variable. For p separate factors, the LR model can be expressed as follows:

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p)}} \quad (3)$$

Here, $P(Y = 1)$ denotes the probability of the presence of Coronary Artery Disease (CAD) and $\beta_0, \beta_1, \dots, \beta_p$ are the regression coefficients. The logistic regression model contains an underlying linear model. The natural logarithm of the ratio of $P(Y = 1)$ to $(1 - P(Y = 1))$ forms a linear model in X_i :

$$g(x) = \ln\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) \quad (4)$$

The function $g(x)$ possesses several advantageous characteristics akin to those of a linear regression model. This feature enables the incorporation of both continuous and categorical variables as independent variables [72,73]. Incorporating interaction variables alongside main effects in the LR model is feasible. Assessing the dataset for observable signs of interaction and confounding terms is crucial in data modeling. A covariate that relates to both the main independent variable and the dependent variable of interest is known as a "confounder". The relationship between the independent and dependent variables is considered to be confounded when associations are detected with both

variables.

To ascertain if a covariate is a confounder in a LR model, a useful approach is to compare the estimated coefficient for the independent variable across models that either include or omit the covariate. If there is any clinically significant change in the estimated coefficient for the independent variable, the covariate functions as a confounding factor and should be included in the model. This guidance holds true irrespective of the statistical significance of the estimated coefficient of the covariate. A systematic method to test for confounders and interactions in LR involves initiating with a main effects model and employing a forward-selection strategy. According to [72], this approach identifies interaction terms that significantly reduce the likelihood ratio test statistic.

3.3. K-Nearest Neighbor Classifier (KNNC)

The coarse to fine KNNC (CFKNNC) employs column vectors X_1, \dots, X_n to represent all N training data. CFKNNC first identifies the training samples most similar to the test sample in a coarse manner. Then, from the n training samples identified, CFKNNC carefully selects $K (K_n)$ closest neighbors of the test sample. Finally, CFKNNC classifies the test sample using the class labels of the K closest neighbors [74].

If X_1, \dots, X_n and the test sample y are not already unit vectors with a length of 1, CFKNNC will transform them into unit vectors beforehand. CFKNNC tries to represent the test sample Y as a linear combination of all training samples. In other words, CFKNNC assumes that the following equation is approximately satisfied:

$$Y = \sum_{i=1}^N Y_i X_i \quad (5)$$

Eq. (5) can be rewritten as

$$Y = X\gamma, \quad (6)$$

Where $X = (X_1, \dots, X_n)$, and $Y = (Y_1, \dots, Y_n)$. To get the answer to Eq. (6), Lagrangian procedure has been used. To ensure that $X\gamma$ closely resembles γ , we aim to minimize the distance between γ and $X\gamma$, denoted as $\|\gamma - X\gamma\|^2$. Furthermore, numerical analysis theories suggest that a small norm for n can lead to a well-generalized solution of Eq. (6). Thus, it is desirable to minimize the norm of γ , $\|\gamma\|^2$. Considering these factors, we define a Lagrangian function $f(\gamma) = \|Y - X\gamma\|^2 + \mu\|\gamma\|^2$, where μ is a positive constant [75]. The optimal solution for Eq. (6) should minimize the function $f(\gamma)$, ensuring that $\frac{\partial f(\gamma)}{\partial \gamma} = 0$ is achieved. $\frac{\partial f(\gamma)}{\partial \gamma} = 0$ indicates that $2(X^T X + \mu I)\gamma = 2X^T Y$. Consequently, $\widehat{\gamma} = (X^T X + \mu I)^{-1} X^T Y$ has been used to find the solution of Eq. (6), where I is the identity matrix. Next, CFKNNC computes:

$$e_i = \|Y - \widehat{\gamma}_i X_i\|^2 \quad (7)$$

The $i - th$ entry of the estimated vector γ , denoted as $\widehat{\gamma}_i$, is represented by $\widehat{\gamma}_i$. The CFKNNC algorithm chooses n training samples with the first n smallest values of e_i , denoted as $Z_1 \dots Z_n$, respectively. These samples are then utilized in a weighted sum to represent the test sample Y .

$$Y = \sum_{i=1}^N W_i Z_i \quad (8)$$

Eq. (8) can be restated as $Y = Zw$, where $w = [w_1 \dots w_n]^T$ and $Z = [Z_1 \dots Z_n]$. The CFKNNC algorithm determines the solution for w using $\widehat{w} = (Z^T Z + \mu I)^{-1} Z^T Y$, where μ is a positive constant and I is the identity matrix. CFKNNC interprets d_i (where d_i represents the $i - th$ entry of d) as the similarity metric between Y and Z_i . A smaller d_i indicates a higher similarity between them. CFKNNC then selects K training samples with the K smallest d_i values from $Z_1 \dots Z_n$ and labels them as $s_1 \dots s_K$, respectively [76]. For every test sample, CFKNNC tallies the number of

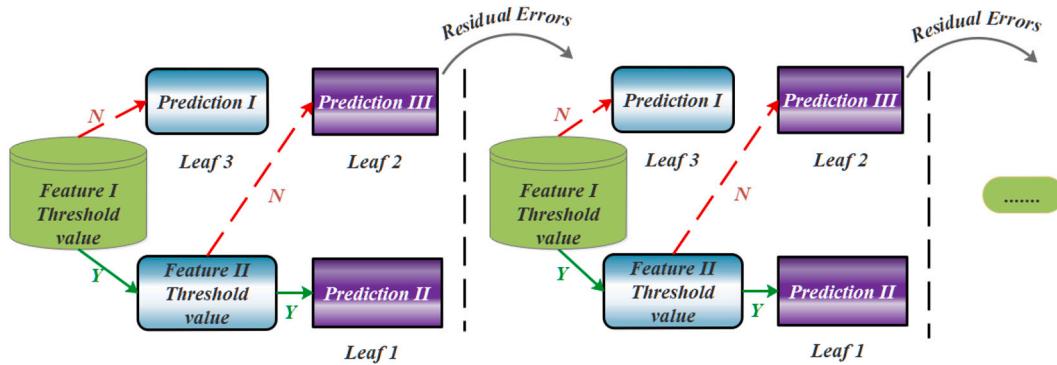


Fig. 3. XGBC structure

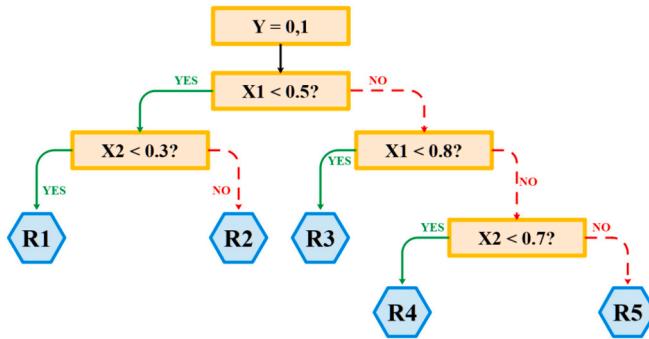


Fig. 4. DTC structure showcasing different aspect of the model.

training samples belonging to the $k - th$ class, indicated as n_k . K denote the total number of classes. If n_k is the highest count among all training samples for different classes (i.e., $n_k \geq n_i$ for all other classes i), CFKNNC assigns the test sample to the $k - th$ class.

3.4. Extreme Gradient Boosting Classifier (XGBC)

This section explores the fundamental theory behind the XGBoost detection algorithm. In terms of computing speed, generalization efficiency, and Versatility, XGBoost represents a significant advancement over the typical Gradient Boosting Decision Tree (GBDT) algorithm. The goal function and optimization method of the XGBoost algorithm are respectively introduced in [77], Eq. (9) determines the target function of XGBoost as follows:

$$F_{\text{Obj}}(\theta) = C(\theta) + \Omega(\theta),$$

where $C(\theta) = c(\hat{y}_i, y_i)$,

$$\Omega(\theta) = \gamma T + \frac{1}{2} \lambda \|w\|^2. \quad (9)$$

The objective function of XGBoost comprises two components: $C(\theta)$ and $\Omega(\theta)$, where θ represents the formula's different parameters. $C(\theta)$ is a differentiable convex loss function used to guide data fitting for the model, measuring the disparity between the actual target y_i and the predicted value \hat{y}_i . The logistic loss $c(\hat{y}_i, y_i) = y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i})$ and mean square loss $c(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$ are conventional eminent loss functions that are repeatedly used. However, $\Omega(\theta)$ serves as a regularization term that penalizes complex models. Here, T represents the number of leaves in the tree, and γ is the learning rate, constrained to 0. Multiplying γ by T facilitates tree pruning, which mitigates overfitting. Unlike traditional GBDT, XGBoost enhances this term with $\frac{1}{2} \lambda \|w\|^2$, where λ is a regularization parameter and w denotes leaf weights. This augmentation further reduces overfitting while enhancing

the model's generalization capabilities.

Despite that, due to the inclusion of model penalty terms with function parameters in the objective function in Eq. (9), conventional optimization techniques encounter challenges. Therefore, it becomes crucial to determine whether the target y_i can be achieved using Eq. (10).

$$C(\theta) = \sum_{i=1}^n c \left(y_i, \hat{y}_i^{(t-1)} + S_t(T_i) + \Omega(\theta) \right). \quad (10)$$

The optimization objective is to construct a tree structure that minimizes the target function in each iteration. To build a current residual regression tree, the tree structure utilizes the findings and residuals from the previous tree (*residuals = real value – predictive value*). The tree generated by instance i in the $t - th$ iteration is denoted as $S_t(T_i)$. As Eq. (9) aims to optimize the square loss function, solving for other loss functions becomes considerably challenging. Consequently, by transforming Eq. (10) through the two order Taylor expansion, Eq. (9) allows for the solution of other loss functions. These include $g_i = \partial_{\hat{y}^{(t-1)}} c(y_i, \hat{y}^{(t-1)})$, $h_i = \partial_{\hat{y}^{(t-1)}}^2 c(y_i, \hat{y}^{(t-1)})$. This also expedites the optimization process because the final objective function depends only on the first and second derivatives of each data point in the error function.

As a result, by employing a second-order Taylor expansion on Eq. (10), Eq. (9) enables the resolution of alternative loss functions. These functions encompass $g_i = \partial_{\hat{y}^{(t-1)}} c(y_i, \hat{y}^{(t-1)})$, $h_i = \partial_{\hat{y}^{(t-1)}}^2 c(y_i, \hat{y}^{(t-1)})$. This approach also accelerates the optimization process, as the ultimate objective function relies solely on the first and second derivatives of each data point within the error function. Fig. 3 illustrates the flowchart of the XGBC.

$$C(\theta) = \sum_{i=1}^n c \left(y_i, \hat{y}_i^{(t-1)} + g_i S_t(T_i) + \frac{1}{2} h_i S_t^2(T_i) \right) + \Omega(\theta). \quad (11)$$

3.5. Decision Tree Classifier (DTC)

Employing a decision tree as a predictive model, decision tree learning connects observations regarding an item with inferences about its target value [78]. The decision tree algorithm, a data mining induction technique, repeatedly divides a dataset of records using either a greedy depth-first or breadth-first approach until each data item is assigned to a specific class [79,80].

The structure of a decision tree comprises internal, root, and leaf nodes arranged in a layout resembling a flowchart. The top node serves as the root, while each internal node denotes a test condition derived from an attribute. Branches illustrate the outcomes of these test conditions, with class labels assigned to each leaf node [81]. The decision tree is built employing a divide-and-conquer approach, ensuring that each path represents a decision rule. Typically, it adopts a greedy strategy, starting from the top and working downwards. The decision tree

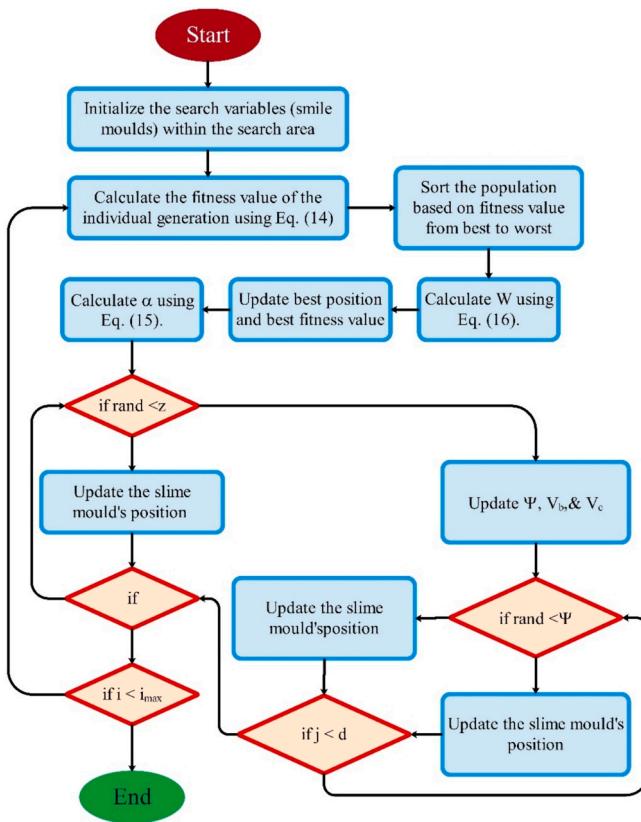


Fig. 5. Process of the SMA

categorization method comprises two phases: tree building and tree pruning. Initially, the tree is constructed using a top-down methodology, iteratively partitioning the data until each item is assigned a class label [82]. This process demands substantial computational resources as it necessitates multiple iterations through the training dataset. Conversely, tree pruning is executed in a bottom-up manner to address overfitting within the tree, thereby enhancing the prediction and classification accuracy of the algorithm. Misclassification errors in decision trees often stem from overfitting. Fig. 4 illustrate structure of the DTC.

3.6. Slime Mold Optimization Algorithm (SMA)

The Slime Mold Algorithm (SMA), inspired by the foraging behavior of slime molds, serves as a metaheuristic algorithm. It encompasses three stages: approach, wrapping, and food collection. Slime molds modify their positions based on the distribution of food. These procedures are characterized mathematically by dedicated equations [61]:

$$\vec{X}^{t+1} = \begin{cases} \text{rand}.(Ub - Lb) + Lb \text{ rand} < z \\ \vec{X}_b^t + \vec{v}_b \cdot (\vec{W} \cdot \vec{X}_A^t - \vec{X}_B^t) \text{ rand} > z, r < k \\ \vec{v}_c \cdot \vec{X}^t \text{ rand} > z, r \geq k \end{cases} \quad (12)$$

Within the slime mold algorithm, \vec{X}^t and \vec{X}^{t+1} represent the current and updated locations, respectively. The slime mold with the highest fitness is designated as \vec{X}_b^t , while \vec{X}_A^t and \vec{X}_B^t represent randomly selected search operators at the $t - th$ iteration. The parameters r and $rand$ indicate random numbers within the range $[0,1]$, and Lb and Ub denote the lower and upper bounds of the search area, respectively. The parameter z signifies a constant value essential for balancing exploration and exploitation. Additionally, parameter k is defined as follows [61]:

$$k = \tanh|F(i) - x_b| \quad (13)$$

In this context, $F(i)$ denotes the fitness of the $i - th$ search agent, and x_b represents the search agent with the most suitable fitness values. In Eq. (14), \vec{v}_c is considered a vector within the interval $[-1,1]$, computed as follows [61]:

$$\vec{v}_c = 1 - \left(\frac{t}{\max_t} \right) \quad (14)$$

Moreover, \vec{v}_b represents a vector within the span $[-a, a]$. The value of a can be obtained as follows [61]:

$$a = \tanh^{-1} \left(\left(\frac{t}{T} \right) + 1 \right) \quad (15)$$

Where T represents the maximum number of repetitions, and W denotes the corresponding weight, as described in [61]:

$$\vec{W}(I) = \begin{cases} 1 + r \times \log \left(\frac{BF - F(i)}{BF - WF} + 1 \right) & \text{condition} \\ 1 - r \times \log \left(\frac{BF - F(i)}{BF - WF} + 1 \right) & \text{others} \end{cases} \quad (16)$$

In this scenario, the term “condition” refers to the fitness level of the individual within the initial segment of the swarm. WF and BF denote the weakest and most optimal fitness values among the searching operators, respectively. Fig. 5 presents the flowchart of the SMA.

3.7. Forest Optimization Algorithm (FOA)

The FOA derives inspiration from the intricate dynamics of forests, which are influenced by a variety of physical, chemical, and biological factors. FOA utilizes a straightforward model to replicate forest dynamics for optimization purposes, considering trees as experimental solutions and their branches as parameters to optimize. Simplifying the process, tree development is governed by three mechanisms: growth, proliferation, and mortality [83].

In the FOA, parameter values are represented by the length of branches. Just as trees in nature constantly adapt their branches to obtain sunlight and nutrients for survival, the FOA employs a growth mechanism in each generation to adjust branch lengths and improve tree vitality. This mechanism, applied to every tree and branch, entails the following steps carried out sequentially [83]:

- Routine for Growth: Begins with a branch initially measuring L_0 within a tree with vitality V_0 . Consequently, the augmentation in the length of this branch is implemented as follows [83]:

$$L = L_0 * (1 + \delta * rand) \quad V > V_0, L \leq L_{max} \quad (17)$$

$$L = L_0 \quad V \leq V_0 \text{ or } L > L_{max}$$

Where L and V represent the new length and vitality of the branch, respectively. δ signifies the predetermined growth speed factor with a value ranging from 0 to 1. Additionally, $rand$ denotes a random value within the $[0,1]$ interval, and L_{max} indicates the maximum length of the branch.

- Procedure for Withering: Similarly, this process is implemented to enhance tree vitality by reducing the length of the branch, as indicated by the following expression [83]:

$$L = L_0 * (1 - \delta * rand) \quad V > V_0, L \geq L_{max} \quad (18)$$

$$L = L_0 \quad V \leq V_0 \text{ or } L < L_{max}$$

This process persists until further reduction in length ceases to improve the tree's vitality or until the length exceeds its limit. The straightforwardness of the growth mechanism becomes apparent. By employing a parameter-sweeping approach, the probability of discovering a local optimum around a trial solution represented by a tree is

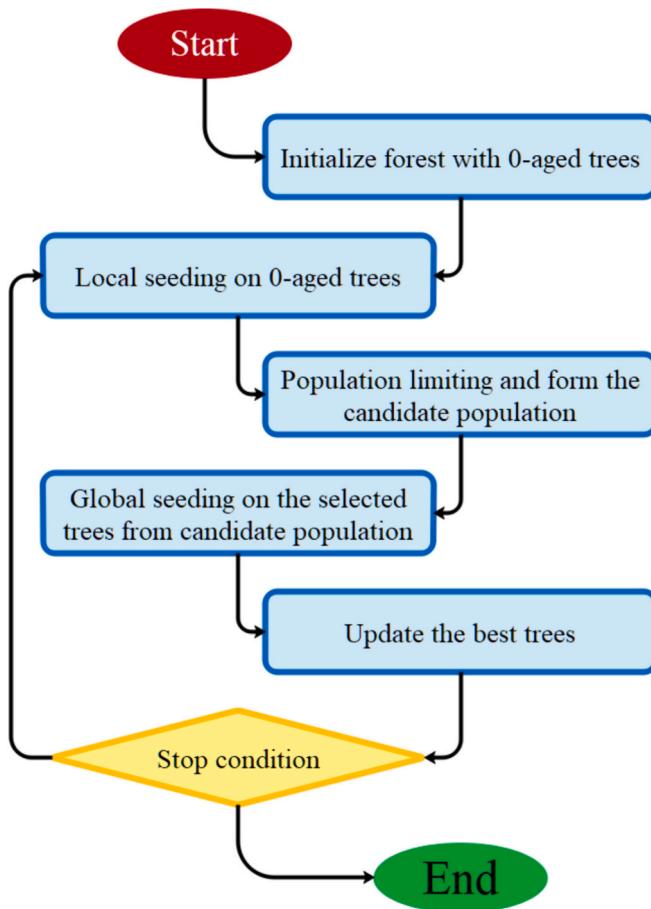


Fig. 6. Flowchart of the FOA [84]

significantly enhanced, thus greatly improving tree vitality. Fig. 6 presents the flowchart of the FOA.

3.8. Pathfinder Algorithm (PFA)

The PFA algorithm draws inspiration from the collective seeking, hunting, and scouting tactics observed in animal teams. Within a team, individuals collaborate to execute tasks, while a leader directs activities such as guiding the group to grazing areas, water sources, and feeding grounds. Leadership roles may rotate based on individuals' aptitudes for achieving objectives [85].

The model allocates positions in 2D, 3D, or D-dimensional spaces, with the most encouraging member assuming the role of the “pathfinder” leader. All solutions are depicted using position vectors, enabling movement across dimensions for effective seeking and tracking. The following model is proposed based on the provided expressions [85]:

$$X(t + \Delta t) = X^0(t) \cdot n + f_i + f_p + \varepsilon \quad (19)$$

In this context, t represents time, X represents position, n is a unit vector disregarding angles, f_i denotes the interaction between pairs of neighbors (X_i, X_j), v refers to the global force associated with the optimal pathfinder location, and ε represents the agent of the vibration vector. The following equation is employed to update the position of the pathfinder [85]:

$$X_P(t + \Delta t) = X_P(t) + \Delta X + A \quad (20)$$

Here, $X_P(t)$ represents the location of the pathfinder, ΔX indicates the displacement of the pathfinder, and A denotes the vector of fluctuation rates. Eqs. (19) and (20) are adjusted into Eq. (21) to investigate the optimal point as the initial modification [85]:

$$X_i^{k+1} = X_i^k + R_1 \cdot (X_j^k - X_i^k) + R_2 \cdot (X_p^k - X_i^k) + \varepsilon, i \geq 2 \quad (21)$$

Where k represents the current iteration, i and j denote the i th and j th members, and R_1 and R_2 indicate the random vectors. The subsequent adjustment can be mathematically represented by Eq. (22) [85]:

$$X_p^{k+1} = X_p^k + 2r \cdot (X_p^k - X_p^{k-1}) + A \quad (22)$$

Here, r represents a randomly selected vector within the $[0,1]$ interval. The values of A and ε can be computed as follows [85]:

$$\varepsilon = \left(1 - \frac{K}{K_{max}}\right) \cdot u_1 \cdot D_{ij} \cdot D_{ij} = \|X_i - X_j\| \quad (23)$$

$$A = u_2 \cdot e^{-\frac{2k}{K_{max}}} \quad (24)$$

In the given equations, u_1 and v represent vectors within the interval $[-1,1]$, D_{ij} denotes the distance between the i -th and j -th members, and K_{max} is the maximum number of iterations for the agent. Further information can be found in [85].

3.9. Giant Armadillo Optimization (GAO)

The GAO method draws inspiration from giant armadillos and serves as a population-based meta-heuristic algorithm. It efficiently solves optimization problems through iterative operations, harnessing the collective search capabilities of its members. Each member of the GAO population, resembling a giant armadillo, contributes to solution identification by determining values for decision variables within the problem space. Mathematically represented as a matrix, these sizable armadillos collectively constitute the algorithm's population. The algorithm begins by randomly assigning their positions in the problem space using Eq. (25).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix}_{n \times m} = \begin{bmatrix} x_{1,1} & \dots & x_{1,d} & \dots & x_{1,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,d} & \dots & x_{i,m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,d} & \dots & x_{n,m} \end{bmatrix}_{n \times m} \quad (25)$$

$$x_{i,d} = lb_d + r_{i,d} \bullet (Ub_d - lb_d) \quad (26)$$

The GAO population matrix is denoted by X , where each i th GAO member (candidate solution) is represented by X_i . A member's search space (decision variable) has d dimensions, which are denoted by d . r is a random number within the interval $[0,1]$, n is the total number of gigantic armadillos, m is the number of decision variables, and lb_d and Ub_d are the decision variable's lower and upper limits, respectively. The location of each enormous armadillo in the problem-solving space indicates a possible solution, making it possible to calculate the value of the objective function for each armadillo. Consequently, the set of evaluated values for the objective function may be represented by Eq. (27).

$$\vec{P} = \begin{bmatrix} P_1 \\ \vdots \\ P_i \\ \vdots \\ P_n \end{bmatrix}_{n \times 1} = \begin{bmatrix} P(X_1) \\ \vdots \\ P(X_i) \\ \vdots \\ P(X_n) \end{bmatrix}_{n \times 1} \quad (27)$$

Here, P_i represents the evaluated objective function for the i th GAO member, and \vec{P} is the vector containing the values of the assessed objective function. The objective function evaluations provide important information on the caliber of possible fixes that the population members have put forth. The member with the lowest value is

considered the poorest, whereas the one with the highest value is regarded as the best. By comparing the updated values of the objective function, the best member is continuously updated alongside the iterative adjustments of the big armadillos' locations inside the area used to solve problems. The position of the greatest Participant obtained throughout the iterations ultimately serves as the algorithm's solution upon completion. The GAO method emulates the hunting strategy of giant armadillos to adjust the placements of population members. This strategy involves attacking termite mounds and subsequently entering them to feed. As a result, each iteration of GAO comprises two phases: exploitation, which mimics digging in termite mounds to locate and consume termites, and exploration, which simulates moving toward termite mounds.

3.9.1. Phase 1: Termite Mound Attack (Phase of Exploration)

GAO's initial step entails revising population members' positions by simulating the hunting behavior of giant armadillos on termite mounds. The algorithm models the dynamic movements used by these large armadillos to search for termite mounds, thereby enhancing its exploration capabilities for global search management. This attack process results in significant changes in the placements of population members.

In the GAO design, each member of the population represents a huge armadillo, and the placements of other individuals with a higher objective function value are compared to termite mounds. Eq. (28) defines the set of potential termite mounds for each individual in the population.

$$TM_i = \{X_k : P_k < P_i \text{ and } k \neq i\}, \text{ where } i = 1, 2, \dots, n \text{ and } k \in \{1, 2, \dots, n\} \quad (28)$$

In this case, P_k indicates the objective function value of the population member X_k , which is greater than the i th gigantic armadillo. The list of potential termite mound locations for the i th armadillo is denoted by TM_i . After picking a termite mound at random, the giant armadillo attacks it. By applying Eq. (29), a new location is determined for every member of the population, simulating the migration of enormous armadillos in the direction of termite mounds. The prior location of the corresponding member is replaced if this new position improves the objective function value as determined by Eq. (30).

$$x_{i,j}^{L1} = x_{i,j} + r_{i,j} \cdot (STM_{i,j} - I_{i,j} \cdot x_{i,j}), \quad (29)$$

$$X_i = \begin{cases} x_i^{L1}, P_i^{L1} \leq P_i, \\ X_i, \text{ else,} \end{cases} \quad (30)$$

For the i th gigantic armadillo, x_i^{L1} denotes the newly determined position from the assaulting phase of the planned GAO, $x_{i,j}^{L1}$ represents its j th dimension, and P_i^{L1} represents the value of its objective function. $I_{i,j}$ is a randomly selected integer, either 1 or 2, and the values $r_{i,j}$ are random numbers within the interval $[0, 1]$. The chosen termite mound for the i th gigantic armadillo is indicated by $STM_{i,j}$.

3.9.2. Phase 2: Termite Mound Excavation (Utilization Phase)

During the updating of population member placements in the problem-solving space is the second part of the GAO. Simulated to mimic giant armadillos burrowing into termite mounds to feed on termites. The positions of these large armadillos undergo slight adjustments as a result of simulating termite-hunting and termite-eating activities. Consequently, the algorithm's capability for managing local searches is enhanced. Eq. (31) determines the new location for each population member by leveraging the expertise of giant armadillos in creating termite mounds. If an improvement in the objective function value is observed, Eq. (32) replaces the original location of the respective member with the new position.

$$x_{i,j}^{L2} = x_{i,j} + (1 - 2r_{i,j}) \cdot \frac{Ub_j - Lb_j}{t} \quad (31)$$

$$X_i = \begin{cases} x_i^{L2}, P_i^{L2} \leq P_i \\ X_i, \text{ else} \end{cases} \quad (32)$$

In this context, P_i^{L2} represents the objective function value, $r_{i,j}$ denotes random integers within the interval $[0, 1]$, t signifies the iteration counter, and x_i^{L2} indicates the new location calculated for the huge armadillo i th, derived from the excavation stage of the proposed GAO. First, the input information is given along with the range of variables and the objective function. Then the population size and the maximum number of repetitions are determined. An initial evaluation population is then generated. Phase one and phase two have been implemented respectively, and finally, by checking the conditions, will find the best solution.

3.10. Metrics

For many ML-related analytical tasks, classifying and predicting incoming data is essential. Statistical metrics like F1-score (F1), Recall (Rec), Accuracy (Acc), Matthews correlation coefficient (MCC) and Precision (Pre) are frequently used to assess the efficacy of classification algorithms. These measurements provide a numerical understanding of the stability and forecast precision of classification systems. By taking into account both false positives and false negatives, the F1-score, for example, offers a fair evaluation. Recall, sometimes referred to as true positive rate or sensitivity, measures how well the model can identify actual positive cases. The model's overall correctness is indicated by accuracy, while its capacity to reduce false positives is indicated by precision, which measures the percentage of correctly predicted positive cases among all instances projected as positive. These metrics can be stated mathematically as follows:

$$F1 = (2 \times \text{Rec} \times \text{Pre}) / (\text{Rec} + \text{Pre}) \quad (33)$$

$$\text{Pre} = \text{Tp} / (\text{Tp} + \text{Fp}) \quad (34)$$

$$\text{Rec} = \text{FpR} = \text{Fp}/\text{P} = \text{Fp} / (\text{Fp} + \text{Fn}) \quad (35)$$

$$\text{Acc} = (\text{Tp} + \text{Tn}) / (\text{Tp} + \text{Tn} + \text{Fp} + \text{Fn}) \quad (36)$$

$$\text{MCC} = \text{tp} \times \text{tn} - \text{fp} \times \text{fn} / \sqrt{(\text{tp} + \text{fp})(\text{tp} + \text{fn})(\text{tn} + \text{fp})(\text{tn} + \text{fn})} \quad (37)$$

Correct rejection occurrences are represented by True Negatives (Tn), whereas accurate forecasts are represented by True Positives (Tp). Missed positive cases and erroneous forecasts are represented by False Positives (Fp) and False Negatives (Fn). When combined, these statistical metrics offer a thorough framework for evaluating the effectiveness of categorization models, which helps ML applications make well-informed decisions.

4. Result and discussion

This section presents a detailed and thorough analysis of heart disease prediction. It utilizes five distinct models, each employing a unique approach to prediction, along with four powerful optimizers to enhance performance. The evaluation process spans across the training, validation, and testing phases, ensuring a comprehensive assessment of each model's effectiveness in predicting heart disease. By encompassing the entire dataset, this analysis aims to provide valuable insights into the comparative performance and suitability of these models for practical applications in heart disease prediction. The array of models has been scrutinized in this analysis encompassed various combinations, including XGBC, XGGA, XGSM, XGPA, XGFO, RFC, RFGA, RFSM, RFPA, RFFO, DTC, DTGA, DTS, DTPA, and DTFO. These models underwent rigorous scrutiny utilizing the metrics delineated in Section 3.10,

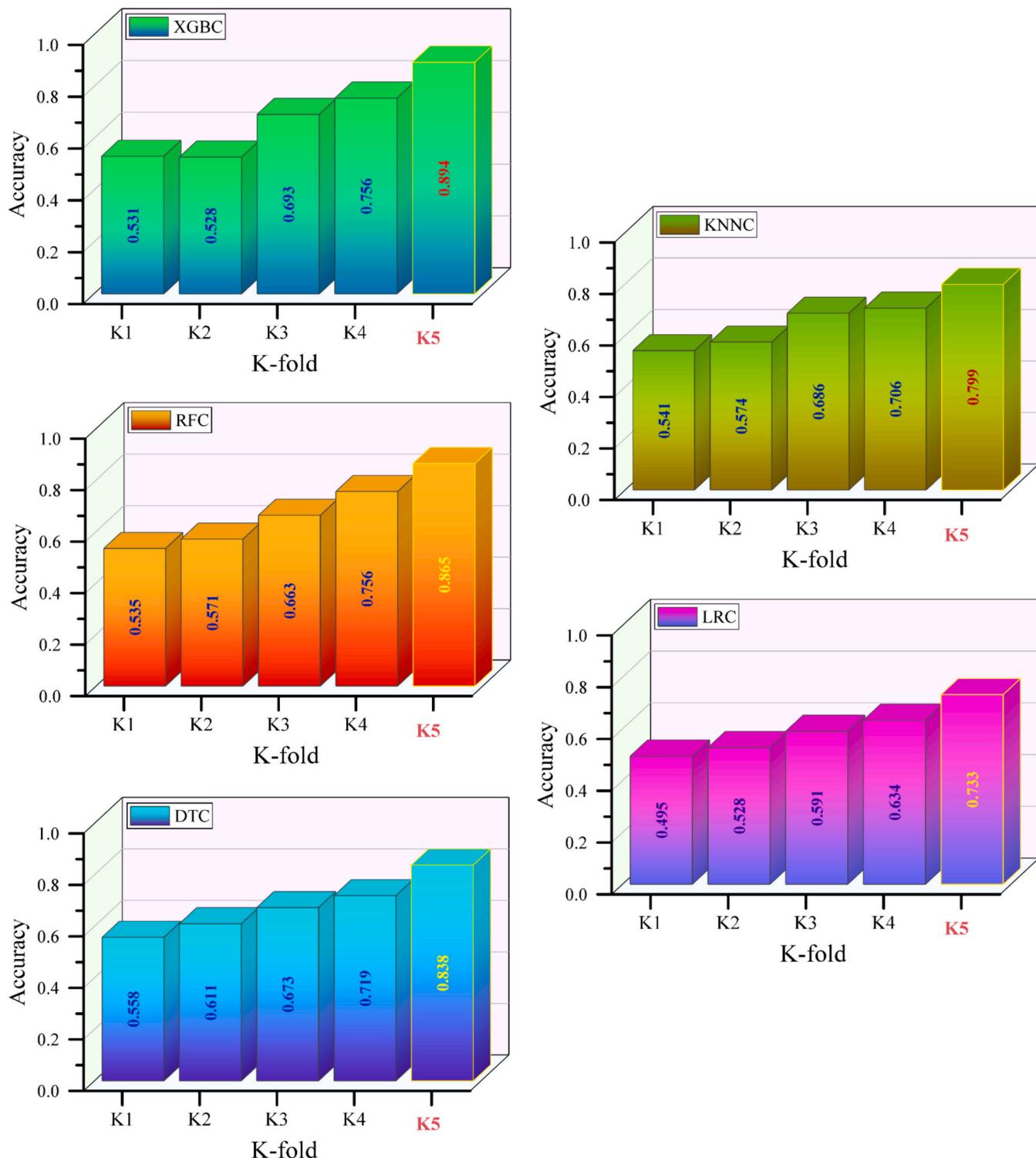


Fig. 7. 3D plot for the results of the k-fold cross validation for the five base models.

facilitating an exhaustive examination of their performance. The overarching aim of this analysis was to furnish a comprehensive and insightful understanding of how these models fare in predicting heart disease, which finally provides solutions that can be very helpful in this field with hybrid structures and models.

The subsequent sections have been incorporated to thoroughly delve into the achieved results: the K-fold cross-validation analysis, the convergence curve assessment, heat maps, investigation performance of models in different conditions, confusion matrixes, and the Taylor

diagram of models. These segments are designed to offer a comprehensive perspective on the findings and their ramifications, thereby enhancing the comprehension of this research endeavor. Since heart diseases are one of the biggest causes of death in the world, early diagnosis can prevent further problems. Through the discovery of these components, a more accurate understanding of the effectiveness and reliability of the models used in the prediction of heart diseases can be achieved.

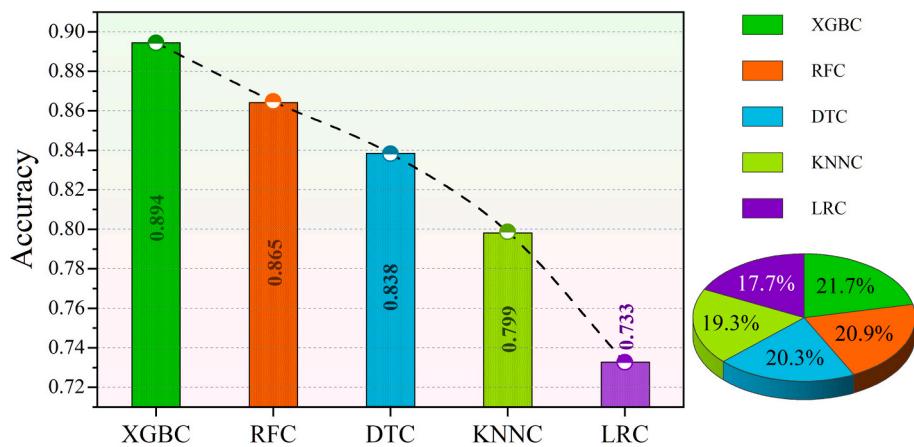


Fig. 8. Column and pie for the difference of the presented models in different folds

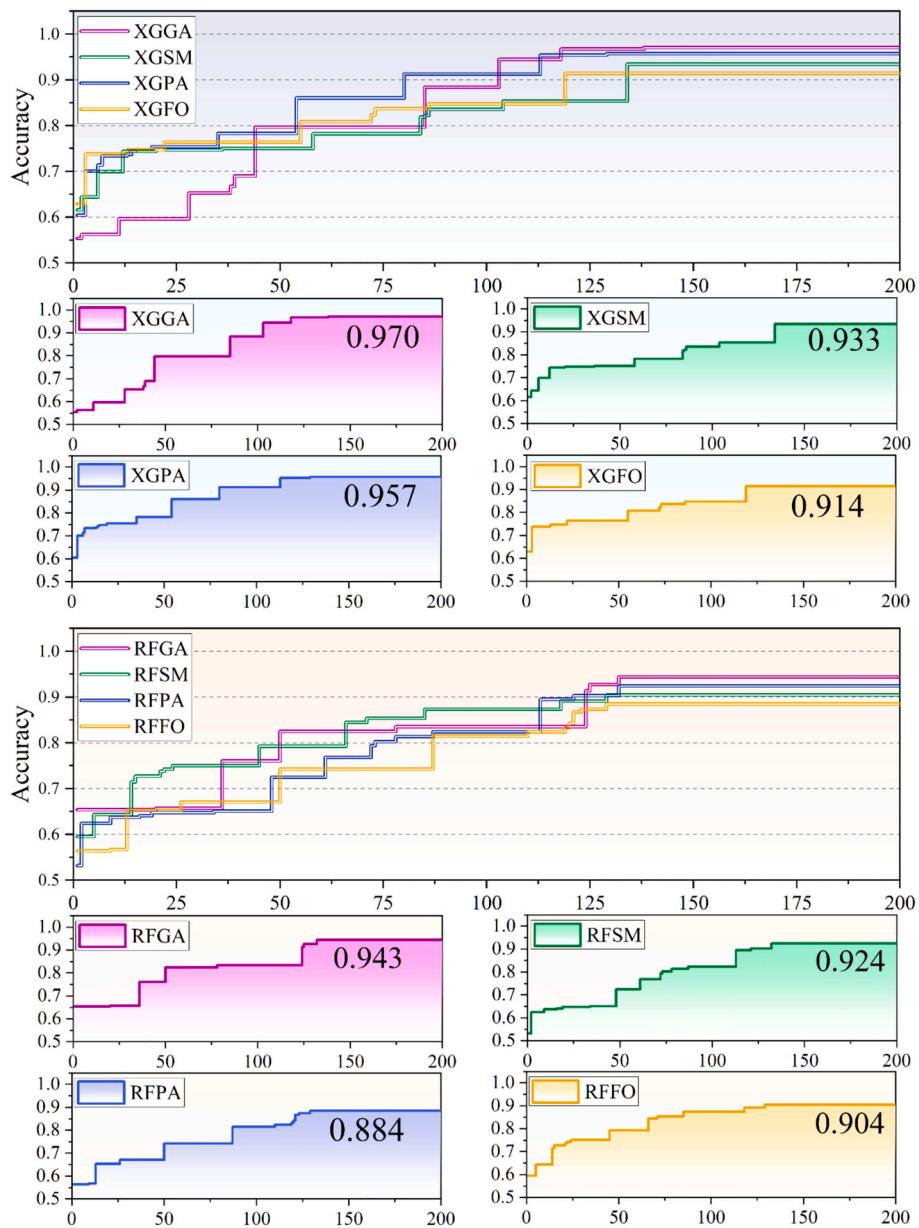


Fig. 9. The convergence curve of the DTC, XGBC, and RFC based hybrid models considering the accuracy value of the models through the iterations.

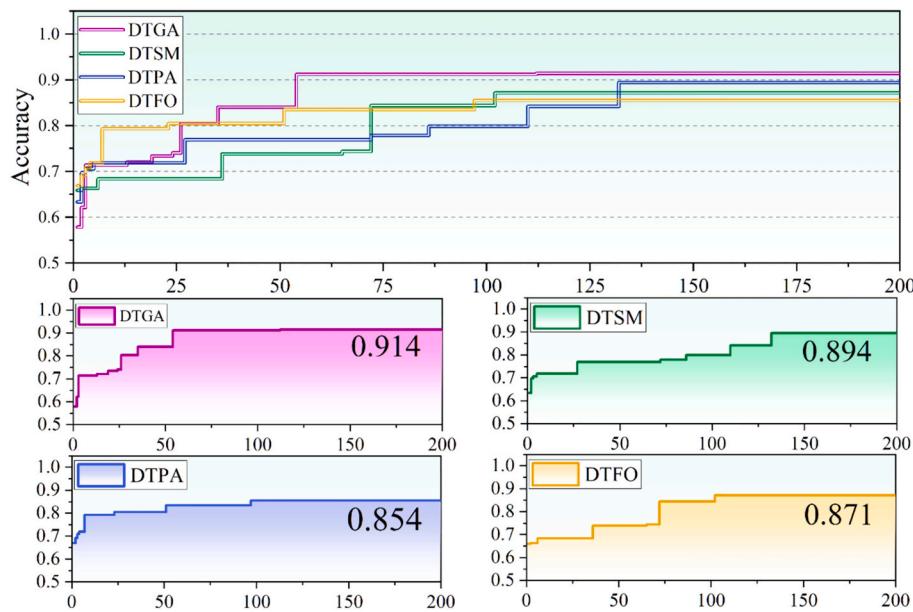


Fig. 9. (continued).

Table 2
The outcomes of both the single and hybrid models of XGBC.

Section	Model	Metrics				
		Accuracy	Precision	Recall	F1_Score	MCC
Training	XGBC	0.901	0.903	0.901	0.900	0.801
	XGGA	0.972	0.972	0.972	0.972	0.943
	XGSM	0.958	0.959	0.958	0.957	0.916
	XGPA	0.962	0.963	0.962	0.962	0.924
	XGFO	0.939	0.941	0.939	0.938	0.879
	XGBC	0.879	0.887	0.879	0.878	0.763
Testing	XGGA	0.967	0.967	0.967	0.967	0.934
	XGSM	0.879	0.879	0.879	0.879	0.757
	XGPA	0.945	0.945	0.945	0.945	0.890
	XGFO	0.857	0.858	0.857	0.857	0.713

4.1. K-fold cross validation

K-fold cross-validation is a widely used technique in machine learning and statistical modeling for evaluating the performance of a predictive model. Its primary goal is to assess how well a model generalizes to unseen data. Unlike a simple train-test split, K-fold cross-validation provides a more robust estimate of the model's performance by evaluating it on multiple subsets of the data, thereby reducing variance in the performance estimate. The process involves partitioning the dataset into K folds or subsets, where K is a predefined number. Common choices for K include 5-fold and 10-fold cross-validation, though other values can be used based on the dataset's size and characteristics. The main objective of K-fold cross-validation is to accurately assess the model's performance and generalization ability while ensuring that each data point is precisely used once during testing. By utilizing multiple subsets of the data for both training and testing, K-fold cross-validation helps to provide a more reliable estimate of the model's performance, reducing potential bias and variance in the evaluation. It also enables better utilization of the available data and helps to identify issues such as overfitting or underfitting by evaluating the model across different subsets of the dataset. In each iteration of K-fold cross-validation, the model is trained on K-1 folds and evaluated on the remaining fold. Fig. 7 presents the results of the k-fold cross-validation. This process is repeated K times, ensuring that every data point contributes to both training and testing. The model's performance is then evaluated using a designated metric in each iteration. Overall, K-fold cross-validation

Table 3
The outcomes of both the single and hybrid models of RFC.

Section	Model	Metrics				
		Accuracy	Precision	Recall	F1_Score	MCC
Training	RFC	0.863	0.869	0.863	0.862	0.728
	RFGA	0.953	0.954	0.953	0.953	0.906
	RFSM	0.920	0.922	0.919	0.919	0.840
	RFPA	0.934	0.935	0.934	0.934	0.867
	RFFO	0.887	0.897	0.887	0.885	0.780
	RFBC	0.868	0.870	0.868	0.867	0.736
Testing	RFGA	0.923	0.923	0.923	0.923	0.845
	RFSM	0.868	0.868	0.868	0.868	0.735
	RFPA	0.901	0.903	0.901	0.901	0.804
	RFFO	0.879	0.879	0.879	0.879	0.757

Table 4
The outcomes of both the single and hybrid models of DTC.

Section	Model	Metrics				
		Accuracy	Precision	Recall	F1_Score	MCC
Training	DTC	0.840	0.847	0.840	0.837	0.681
	DTGA	0.926	0.927	0.926	0.926	0.853
	DTSM	0.892	0.896	0.892	0.890	0.785
	DTPA	0.910	0.911	0.910	0.910	0.819
	DTFO	0.868	0.875	0.868	0.866	0.738
	DTBC	0.835	0.838	0.835	0.834	0.670
Testing	DTGA	0.868	0.869	0.868	0.868	0.736
	DTSM	0.824	0.824	0.824	0.824	0.646
	DTPA	0.857	0.859	0.857	0.857	0.716
	DTFO	0.824	0.826	0.824	0.823	0.646

offers a more comprehensive evaluation of a model's performance by utilizing the entire dataset for training and testing, as shown in Fig. 8. For all models in 5-fold, has the highest accuracy. However, XGBC has the highest accuracy with 0.894 and LRC has the lowest accuracy with 0.733.

Also, Fig. 8 represents the column and pie display of the different models. Due to these results, the KNNC and LRC model in the following are to be left out because their accuracy is low.

4.2. Convergence curve

A convergence curve depicts how a particular metric or objective

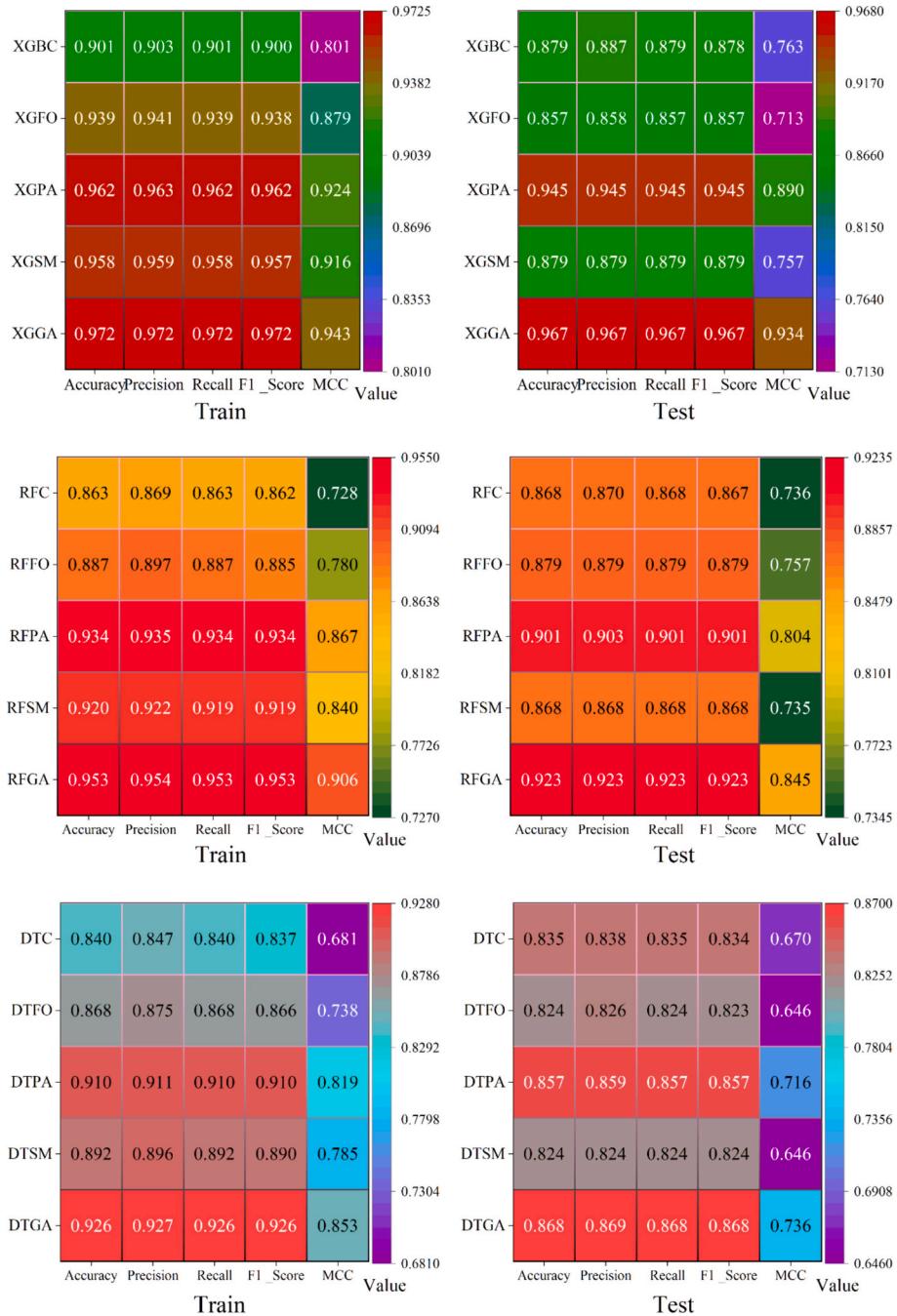


Fig. 10. Heat map for the performance of the models in train and test phases considering their metric values.

function evolves across iterations or epochs during model training. It signifies “convergence,” a point where the model’s performance stabilizes or levels off, suggesting that additional training iterations are unlikely to substantially enhance the model’s performance. This curve usually charts the metric values (like loss function, accuracy, or validation error) against the iteration or epoch count. A convergence curve offers valuable insights into how a machine learning model is trained that it’s very useful for detecting best model, especially when there are many models, these diagrams are good helper for analyze models performance. Fig. 9 illustrates the convergence curve for hybrid models based on three selected models, that’s mean XGBC, RFC, and DTC. In this curve first shows the all hybrid model’s convergence curves in one diagram, and in the next step, the related convergence curve for each hybrid model is shown in a single diagram.

For XGBC-based hybrid models, XGGA with 0.970 has the highest accuracy, XGPA with the accuracy of 0.957, xgsm with the accuracy of 0.933, and XGFO with the accuracy of 0.914 has the lowest values, respectively. For RFC-based hybrid models, RFGA with 0.943 has the highest accuracy, RFSM with the accuracy of 0.924, RFFO with the accuracy of 0.904, and RFPA with the accuracy of 0.884 has the lowest values, respectively. Finally, For DTC-based hybrid models, DTGA with 0.914 has the highest accuracy, DTSM with the accuracy of 0.894, DTPA with the accuracy of 0.854, and DTFO with the accuracy of 0.871 has the lowest values, respectively. According to the results, it is quite evident that in all three models based on Giant Armadillo optimization, the measured accuracy has the highest value.

Table 2 shows the single and hybrid models of XGBC for all metrics mentioned. This table delineates the outcomes derived from the

Table 5
Model performance in the two different conditions.

Model	Condition	Metric		
		precision	recall	f1-Score
XGBC	Alive	0.934	0.826	0.877
	Die	0.867	0.952	0.9750
XGGA	Alive	0.985	0.949	0.967
	Die	0.959	0.988	0.973
XGSM	Alive	0.954	0.899	0.925
	Die	0.919	0.964	0.941
XGPA	Alive	0.963	0.942	0.952
	Die	0.952	0.970	0.961
XGFO	Alive	0.931	0.877	0.903
	Die	0.902	0.946	0.923
RFC	Alive	0.908	0.783	0.841
	Die	0.837	0.933	0.883
RFGA	Alive	0.962	0.913	0.937
	Die	0.930	0.970	0.950
RFSM	Alive	0.929	0.855	0.891
	Die	0.886	0.946	0.915
RFPA	Alive	0.926	0.906	0.916
	Die	0.923	0.939	0.931
RFFO	Alive	0.933	0.804	0.864
	Die	0.853	0.952	0.900
DTC	Alive	0.887	0.739	0.806
	Die	0.809	0.921	0.861
DTGA	Alive	0.918	0.891	0.904
	Die	0.911	0.933	0.922
DTSM	Alive	0.896	0.812	0.852
	Die	0.854	0.921	0.886
DTPA	Alive	0.884	0.884	0.884
	Die	0.903	0.903	0.903
DTFO	Alive	0.898	0.768	0.828
	Die	0.827	0.927	0.874

evaluation of both the single and hybrid models of XGBC across five pivotal performance metrics, namely Accuracy, Precision, Recall, F1-Score, and MCC. Notably, the XGGA model emerges as the frontrunner, showcasing superior values across all indexes during both the testing and training phases. Specifically, the aforementioned metrics attained remarkable scores of 0.972 for Accuracy, 0.972 for Precision, 0.972 for Recall, 0.972 for F1-Score, and 0.943 for MCC, underscoring the robustness and efficacy of the XGGA model in comparison to its counterparts. [Tables 3](#) shows the single and hybrid models of RFC for all metrics mentioned. Similar to [Table 2](#), this table presents the results obtained from assessing both the single and hybrid models of RFC across five crucial performance metrics: Accuracy, Precision, Recall, F1-Score, and MCC. The RFGA hybrid model demonstrates superior performance, exhibiting the highest values for all five metrics. Specifically, these metrics achieved notable scores of 0.953 for Accuracy, 0.954 for Precision, 0.953 for Recall, 0.953 for F1-Score, and 0.906 for MCC. These results underscore the robustness and effectiveness of the RFGA model when compared to alternative models. Finally, [Table 4](#) also shows the results obtained for RFC and DTC models, respectively. In alignment with earlier tables, the current one showcases the peak values for Accuracy, Precision, Recall, F1-Score, and MCC within the DTGA hybrid model. This model showcases superior performance, displaying the highest scores across all five metrics. Specifically, these metrics attained noteworthy scores of 0.926 for Accuracy, 0.927 for Precision, 0.926 for Recall, 0.927 for F1-Score, and 0.853 for MCC. These findings emphasize the resilience and efficacy of the DTGA model when contrasted with other models.

A heatmap for the performance of models is a graphical representation that uses color-coded cells to visualize the performance of different models across various metrics or datasets. Each cell in the heatmap corresponds to a combination of a model and a metric or dataset, and the color of the cell indicates the performance of the model for that metric or dataset. [Fig. 10](#) shows the heat map for the performance of the models in the train and test phases. This figure shows the heat map for all three models, XGBC, RFC and DTC, each according to

the mentioned metrics. For example, as the best model for review, the XGGA model has been selected to review the results shown in its heatmap. The results are calculated for two modes, test and train, and are shown in separate figures. The values shown for the five key parameters are exactly the values in table two, which can also be used to compare one hand with other models. Next to each of these figures is a colored bar, which has different colors according to the values obtained for different parameters. Based on this, the XGGA model has a brighter red color in most of the parameters, according to this color bar, it has the best values for the parameters.

The [Table 5](#) examines the performance of each of the models in two different modes and for three metrics. As can be seen, the XGGA model has the highest values for all three specified metrics. This model has provided Precision values equal to 0.985 for Alive mode, 0.988 for Recall and 0.967 for f1-score and Precision values equal to 0.985 for Die mode, 0.988 for Recall and 0.973 for f1-score, which are the highest values among all models. After this model, XGPA has the best values. For Alive three key parameters, precision, recall, and f1-Score obtained 0.963, 0.942, and 0.952, respectively. However, for the Die three key parameters, precision, recall, and f1-Score obtained 0.952, 0.970, and 0.961, respectively. In the RFC-based models, RFGA obtained the best values with Alive three key parameters, precision, recall, and f1-Score obtained 0.962, 0.913, and 0.937, respectively, and for the Die three key parameters, precision, recall, and f1-Score obtained 0.930, 0.970, and 0.950, respectively. It can be inferred from this table we can get the fact that generally, XGBC-based models show better results.

[Fig. 11](#) shows the line symbol plots for all three base models as well as models based on them. In each of these figures, a measured value is given for both live and dead states. At the same time, the performance of each of these models was examined both individually and hybridly. The results obtained for the XGBC-based models show that the XGGA model has the best performance among all the XGBC-based models with the lowest amount of difference with the measured value. This model has a difference of less than 5.5 % for the live state and less than 1.2 % for the dead state compared to the measured value, which is considered an excellent value. For the RFC-based models, the RFGA model has the best value compared to the RFC-based models, with a difference of approximately 8.6 % for the live state and a difference of approximately 3 % for the dead state. Also, for the DTC-based models, the DTGA model has the best value compared to the DTC-based models, with a difference of approximately 10.8 % for the live state and a difference of approximately 6.6 % for the dead state. In [Fig. 12](#), the confusion matrix is drawn for the accuracy of the models, and the analysis of these values was mentioned as a percentage. As the best model that has obtained good results compared to all other models. The XGGA model is chosen for investigation. According to the confusion matrix shown for this model, it can be seen that out of 138 live items measured, this model has identified 131 items as alive, which is correct, and 7 items as dead, which is incorrect. For the dead state, the results of this model are extraordinary. Out of 165 measured dead cases, it detected only 2 live cases, which is incorrect, but at the same time, it detected 163 dead cases, which is correct. A similar argument can be made for the rest of the confusion matrices.

A Taylor diagram is a visual tool utilized to compare various variables or models with a reference model. It illustrates the correlation, root mean square error (RMSE), and standard deviation of different models or datasets in relation to a reference model. In the realm of machine learning, Taylor diagrams can be customized to assess how well different models perform based on their correlation, RMSE, and standard deviation compared to a reference model or dataset. [Fig. 13](#) shows the Taylor diagram related to all three selected models along with their hybrid models. In all three diagrams, the hybrid models based on Armadillo have the best results, and in general, the XGGA model has obtained the best value.

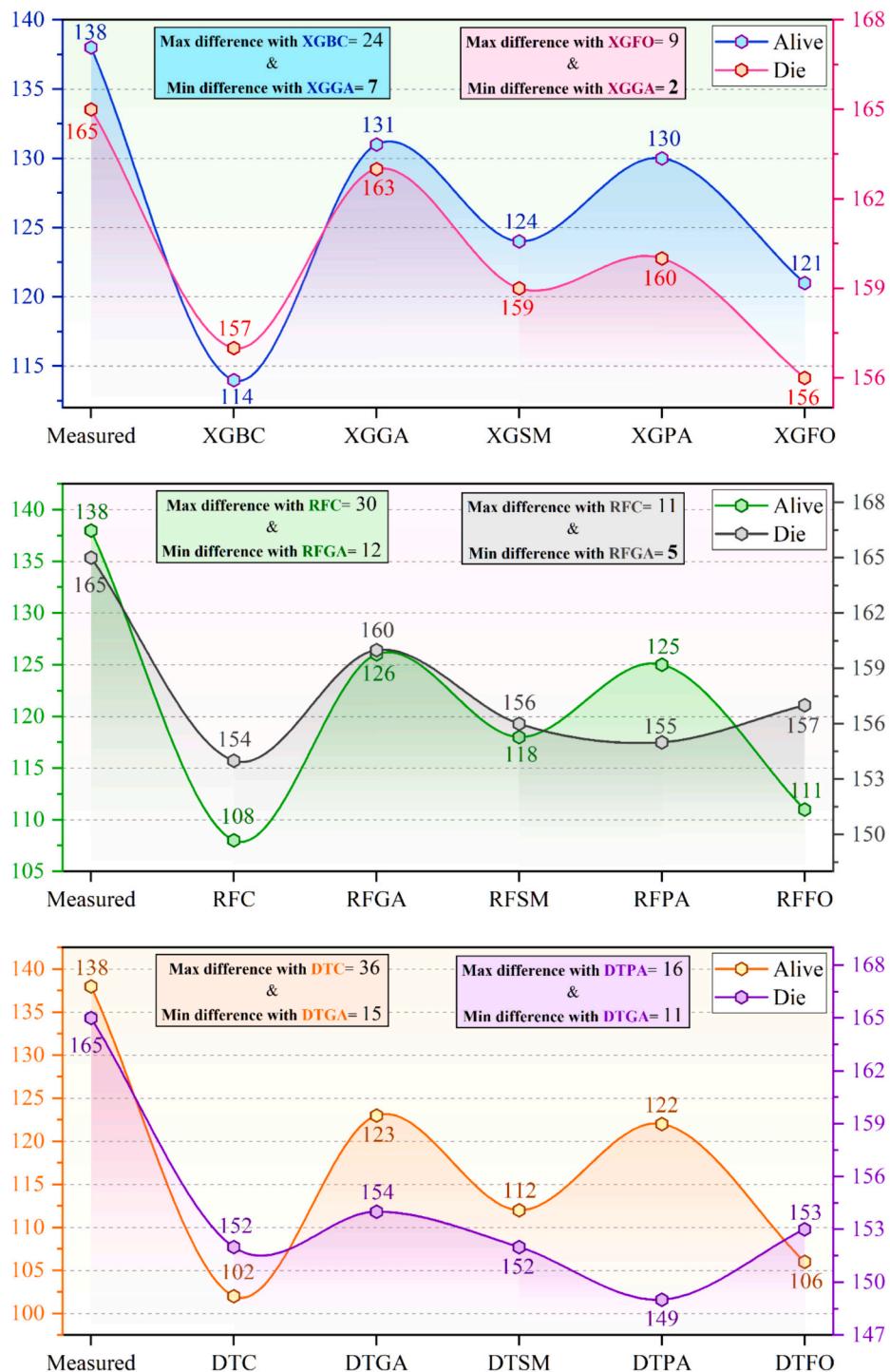


Fig. 11. Line symbol plot for the difference percentage of the models predicted and measured values.

5. Comparison with existing methodologies

Table 6 presents a comparison of the XGGA model's accuracy with several existing methods used for heart disease prediction. The accuracy scores reflect the proportion of correctly predicted instances out of the total number of instances in the dataset. The XGGA model achieved an accuracy of 0.972, demonstrating its high effectiveness in predicting heart disease compared to other models. In comparison, the Random Forest (RF) model reported an accuracy of 0.885, which, while robust, is notably lower than that of the XGGA. The Xgboost model achieved an accuracy of 0.836, indicating strong performance but still falling short

compared to the XGGA. The K-Nearest Neighbors (KNN) model achieved an accuracy of 0.875, showing good performance but not surpassing the XGGA's accuracy. Another variant of the KNN model reached an accuracy of 0.868, which, although effective, is still lower than the XGGA.

This comparison underscores the superior performance of the XGGA model, with its accuracy surpassing that of other widely used methods. The high accuracy of XGGA highlights its potential for effective heart disease prediction, leveraging advanced optimization and ensemble techniques.

	Alive	Die		Alive	Die
Alive	114	24	Alive	131	7
Die	8	157	Die	2	163
XGBC					
	Alive	Die		Alive	Die
Alive	124	14	Alive	130	8
Die	6	159	Die	5	160
XGSM					
	Alive	Die		Alive	Die
Alive	121	17	Alive	108	30
Die	9	156	Die	11	154
XGFO					
	Alive	Die		Alive	Die
Alive	126	12	Alive	118	20
Die	5	160	Die	9	156
RFGA					
	Alive	Die		Alive	Die
Alive	125	13	Alive	111	27
Die	10	155	Die	8	157
RFPA					
	Alive	Die		Alive	Die
Alive	102	36	Alive	123	15
Die	13	152	Die	11	154
DTC					
	Alive	Die		Alive	Die
Alive	131	7	Alive	123	15
Die	2	163	Die	11	154
XGGA					
	Alive	Die		Alive	Die
Alive	130	8	Alive	108	30
Die	5	160	Die	11	154
XGPA					
	Alive	Die		Alive	Die
Alive	108	30	Alive	118	20
Die	11	154	Die	9	156
RFC					
	Alive	Die		Alive	Die
Alive	118	20	Alive	111	27
Die	9	156	Die	8	157
RFSM					
	Alive	Die		Alive	Die
Alive	111	27	Alive	123	15
Die	8	157	Die	11	154
RFFO					
	Alive	Die		Alive	Die
Alive	111	27	Alive	123	15
Die	8	157	Die	11	154
DTGA					
	Alive	Die		Alive	Die
Alive	123	15	Alive	123	15
Die	11	154	Die	11	154

Fig. 12. Confusion matrix for the accuracy of the models in four presented conditions.

	Alive	Die		Alive	Die
Alive	112	26		122	16
Die	13	152		16	149
DTSM			DTPA		
	Alive	Die		Alive	Die
Alive	106	32		106	32
Die	12	153		12	153
DTFO					

Fig. 12. (continued).

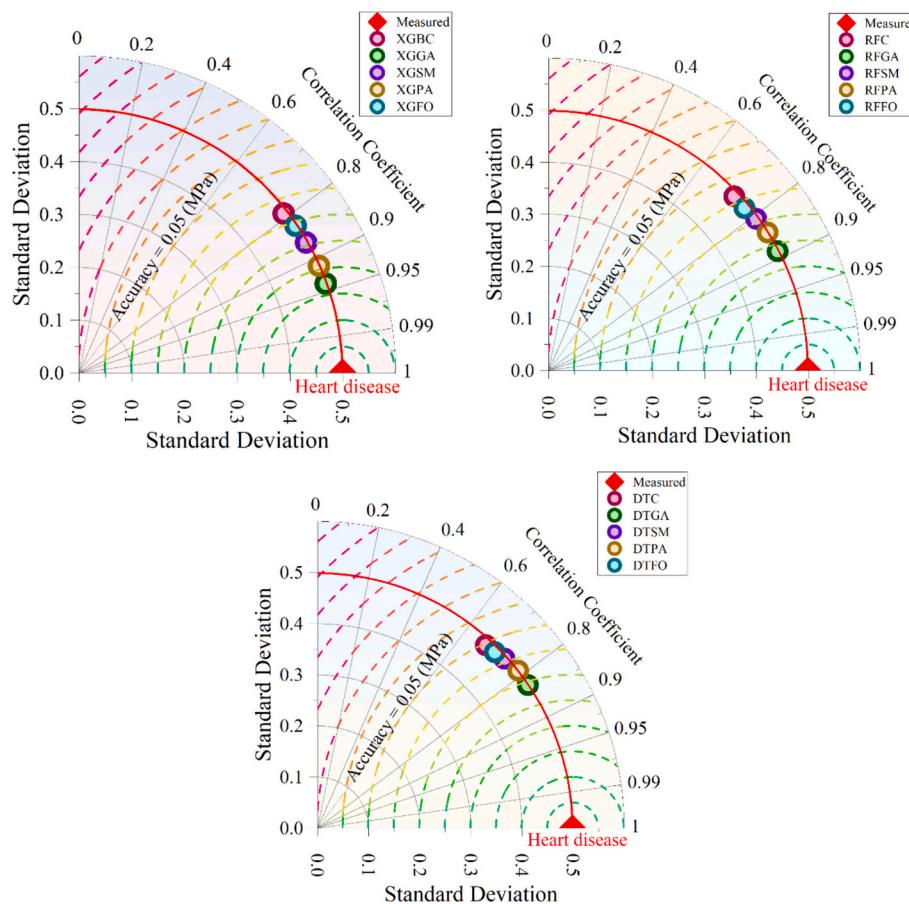


Fig. 13. Taylor diagram of the presented single and hybrid models using the heart disease measured value as the reference point.

6. Conclusion

In this article, the heart disease prediction process was investigated using five machine learning algorithms RFC and LRC, KNNC, XGBC, and DTC models along with four very powerful optimizers GAO, FOA, and

SMA. In addition, other things such as the assembly of models were also examined. Due to the poor results obtained by two KNNC and LRC models, the analysis of these two models was omitted in the further work. First, all the methods were examined from a theoretical point of view to create mental preparation for their use. Then, considering other

Table 6

Comparison of the XGGA with the existing methods.

Model	Metric
	Accuracy
XGGA	0.972
RF [86]	0.885
Xgboost [86]	0.836
KNN [87]	0.875
KNN [88]	0.868

parameters, the desired features were selected. By combining XGBC, RFC, and DTC models with GAO, FOA, PFA and SMA optimizers, a total of fifteen single and hybrid models were introduced, including XGB, XGTO, XGCM, XGSB, XGBC, XGGA, XGSM, XGPA, XGFO, RFC, RFGA, RFSM, RFPA, RFFO, DTC, DTGA, DTSM, DTPA, and DTFO. To check the performance of these models, five key parameters mentioned below have been used. The results were calculated for index values including F1 Score, Recall, Precision, Accuracy, and MCC. These values provide a very comprehensive and accurate view of the performance of all models. In addition, for all models, the visualization view in three parts of training, validation and testing was calculated and shown graphically, which is very useful in comparing the models with each other. By examining the obtained results, it was found that among the single models, XGBC has better performance than the others. Among the hybrid models, the XGGA model showed the best performance. The values acquired for the five specified parameters of the XGGA model are as follows: Accuracy stands at 0.972, Precision measures 0.972, Recall equals 0.972, F1-Score registers at 0.972, and MCC reaches 0.943. These metrics collectively showcase the remarkably high accuracy and performance of the XGGA model. When juxtaposed against alternative models, it becomes evident that the XGGA model outshines its counterparts, underscoring its effectiveness and reliability in predictive tasks. By comparing the results obtained for all models, it can be seen that the XGGA model has obtained the best results, and other diagrams such as the confusion matrix and the Taylor diagram also confirm this fact. Each of these graphs and figures emphasize the results obtained in the superiority of the XGGA model.

Ethical approval

N/A (Not Applicable).

CRediT authorship contribution statement

Haifeng Zhang: Data curation, Conceptualization, Formal analysis, Software, Visualization, Writing – original draft. **Rui Mu:** Project administration, Funding acquisition, Investigation, Methodology, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

None.

Data availability

Data will be made available on request.

Acknowledgements

Item number:2022-2-33Y

Project name: Curative effect of new mode of cardiac rehabilitation combined with Guanxinshutong capsule on patients with ischemic cardiomyopathy.

References

- D. Chicco, G. Jurman, Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone, BMC Med. Inform. Decis. Mak. 20 (1) (Dec. 2020) 16, <https://doi.org/10.1186/s12911-020-1023-5>.
- R. Valarmathi, T. Sheela, Heart disease prediction using hyper parameter optimization (HPO) tuning, Biomed. Signal Proc. Control 70 (Sep. 2021) 103033, <https://doi.org/10.1016/j.bspc.2021.103033>.
- D. Cenitta, R.V. Arjunan, K.V. Prema, Ischemic heart disease multiple imputation technique using machine learning algorithm, Eng. Sci. 19 (2022) 262–272.
- S. Giannitsi, M. Bougiaki, A. Bechlioulis, A. Kotsia, L.K. Michalis, K.K. Naka, 6-minute walking test: a useful tool in the management of heart failure patients, Ther. Adv. Cardiovasc. Dis. 13 (Jan. 2019), <https://doi.org/10.1177/1753944719870084>, p. 1753944719870084.
- H.E.S. Alsafi, O.N. Ocan, A novel intelligent machine learning system for coronary heart disease diagnosis, Appl. Nanosci. (2021) 1–8.
- T. Jaarsma, et al., Self-care of heart failure patients: practical management recommendations from the heart failure Association of the European Society of cardiology, Eur. J. Heart Fail. 23 (1) (Jan. 2021) 157–174, <https://doi.org/10.1002/ejhf.2008>.
- A. Groenewegen, F.H. Rutten, A. Mosterd, A.W. Hoes, Epidemiology of heart failure, Eur. J. Heart Fail. 22 (8) (Aug. 2020) 1342–1356, <https://doi.org/10.1002/ejhf.1858>.
- L. Hersberger, et al., Individualized nutritional support for hospitalized patients with chronic heart failure, J. Am. Coll. Cardiol. 77 (18) (May 2021) 2307–2319, <https://doi.org/10.1016/j.jacc.2021.03.232>.
- Z. Li, H. Zhao, J. Wang, Metabolism and chronic inflammation: the links between chronic heart failure and comorbidities, Front. Cardiovasc. Med. 8 (2021) 650278.
- Y. Kumar, et al., Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things, Wirel. Commun. Mob. Comput. 2021 (Dec. 2021) 1–16, <https://doi.org/10.1155/2021/1616725>.
- Y. Jiang, et al., The effectiveness of a nurse-led home-based heart failure self-management programme (the HOM-HEMP) for patients with chronic heart failure: a three-arm stratified randomized controlled trial, Int. J. Nurs. Stud. 122 (Oct. 2021) 104026, <https://doi.org/10.1016/j.ijnurstu.2021.104026>.
- H. Jindal, S. Agrawal, R. Khera, R. Jain, P. Nagrath, Heart disease prediction using machine learning algorithms, IOP Conf. Ser. Mater. Sci. Eng. 1022 (1) (2021) 012072, <https://doi.org/10.1088/1757-899X/1022/1/012072>.
- G. Bazoukis, et al., Machine learning versus conventional clinical methods in guiding management of heart failure patients—a systematic review, Heart Fail. Rev. 26 (1) (Jan. 2021) 23–34, <https://doi.org/10.1007/s10741-020-10007-3>.
- P. Rani, R. Kumar, N.M.O.S. Ahmed, A. Jain, A decision support system for heart disease prediction based upon machine learning, J. Reliab. Intell. Environ. 7 (3) (Sep. 2021) 263–275, <https://doi.org/10.1007/s40860-021-00133-6>.
- H.V. Ramachandra, P. Chavan, A. Ali, H.C. Ramaprasad, Ensemble machine learning techniques for pancreatic cancer detection, in: In 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), IEEE, 2023, pp. 1–5.
- A. Ajil, A. Ali, H.V. Ramachandra, T.A. Nadaf, Enhancing the healthcare by an automated detection method for PCOS using supervised machine learning algorithm, in: In 2023 International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS), IEEE, 2023, pp. 166–170.
- S. Jahan, et al., Automated invasive cervical cancer disease detection at early stage through suitable machine learning model, SN Appl. Sci. 3 (10) (2021) 806, <https://doi.org/10.1007/s42452-021-04786-z>.
- N. Ahmed, et al., Machine learning based diabetes prediction and development of smart web application, Int. J. Cognitive Comp. Eng. 2 (2021) 229–241, <https://doi.org/10.1016/j.ijcce.2021.12.001>.
- M. Kavitha, G. Gnaneshwar, R. Dinesh, Y.R. Sai, R.S. Suraj, Heart disease prediction using hybrid machine learning model, in: 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 1329–1333, <https://doi.org/10.1109/ICICT50816.2021.9358597>.
- A. Ali, V.R. Hullipalem, S.S. Patil, R. Adbulkader, Consensus pattern selection from structured profile using multiobjective algorithm, Int. J. Adv. Sci. Technol. 28 (8) (2019) 294–305.
- N.C. Staub, Pulmonary edema, Physiol. Rev. 54 (3) (Jul. 1974) 678–811, <https://doi.org/10.1152/physrev.1974.54.3.678>.
- M.M. Ali, et al., Machine learning-based statistical analysis for early stage detection of cervical cancer, Comput. Biol. Med. 139 (2021) 104985, <https://doi.org/10.1016/j.combiomed.2021.104985>.
- M.H. Rooman, U. Bose, M.R. Hasan, M.S. Arman, B.K. Paul, Identification of drugs and association of genes with diseases among anxiety, bipolar disorder, heart disease and stress: A computational biology approach, in: In 2023 26th International Conference on Computer and Information Technology (ICCIT), IEEE, 2023, pp. 1–6.
- G. Dougherty, Medical Image Processing: Techniques and Applications, Springer Science & Business Media, 2011.
- S. Bhattacharya, et al., Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey, Sustain. Cities Soc. 65 (Feb. 2021) 102589, <https://doi.org/10.1016/j.scs.2020.102589>.
- X. Liu, L. Song, S. Liu, Y. Zhang, A review of deep-learning-based medical image segmentation methods, Sustainability 13 (3) (Jan. 2021) 1224, <https://doi.org/10.3390/su13031224>.
- A. Reinke et al., “Common limitations of image processing metrics: A picture story,” arXiv preprint [arXiv:2104.05642](https://arxiv.org/abs/2104.05642), 2021.

- [28] M.M. Ali, B.K. Paul, K. Ahmed, F.M. Bui, J.M.W. Quinn, M.A. Moni, Heart disease prediction using supervised machine learning algorithms: performance analysis and comparison, *Comput. Biol. Med.* 136 (2021) 104672, <https://doi.org/10.1016/j.combiomed.2021.104672>.
- [29] F. Poustchi, et al., Combination therapy of killing diseases by injectable hydrogels: from concept to medical applications, *Adv. Healthc. Mater.* 10 (3) (Feb. 2021), <https://doi.org/10.1002/adhm.202001571>.
- [30] J. Jose, et al., An image quality enhancement scheme employing adolescent identity search algorithm in the NSST domain for multimodal medical image fusion, *Bimed. Signal Proc. Control* 66 (Apr. 2021) 102480, <https://doi.org/10.1016/j.bspc.2021.102480>.
- [31] A. Garg, V. Mago, Role of machine learning in medical research: a survey, *Comput. Sci. Rev.* 40 (May 2021) 100370, <https://doi.org/10.1016/j.cosrev.2021.100370>.
- [32] D.R. Sarvamangala, R.V. Kulkarni, Convolutional neural networks in medical image understanding: a survey, *Evol. Intell.* 15 (1) (Mar. 2022) 1–22, <https://doi.org/10.1007/s12065-020-00540-3>.
- [33] M. Pal, S. Parija, Prediction of heart diseases using random forest, *J. Phys. Conf. Ser.* (2021) 012009. IOP Publishing.
- [34] G.S. Sajja, M. Mustafa, K. Phasinam, K. Kaliyaperumal, R.J.M. Ventayen, T. Kassanuk, Towards application of machine learning in classification and prediction of heart disease, in: 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), 2021, pp. 1664–1669, <https://doi.org/10.1109/ICESC51422.2021.9532940>.
- [35] C. Bemandi, E. Miranda, M. Aryuni, Machine-learning-based prediction models of coronary heart disease using naïve bayes and random forest algorithms, in: 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), 2021, pp. 232–237, <https://doi.org/10.1109/ICSECS-ICOCSIM52883.2021.00049>.
- [36] M. Wang, X. Yao, Y. Chen, An imbalanced-data processing algorithm for the prediction of heart attack in stroke patients, *IEEE Access* 9 (2021) 25394–25404, <https://doi.org/10.1109/ACCESS.2021.3057693>.
- [37] M.G. El-Shafiey, A. Hagag, E.-S.A. El-Dahshan, M.A. Ismail, A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest, *Multimed. Tools Appl.* 81 (13) (May 2022) 18155–18179, <https://doi.org/10.1007/s11042-022-12425-x>.
- [38] R.R. Sanni, H.S. Guruprasad, Analysis of performance metrics of heart failed patients using Python and machine learning algorithms, *Global Trans. Proc.* 2 (2) (Nov. 2021) 233–237, <https://doi.org/10.1016/j.gtplp.2021.08.028>.
- [39] E.M. Senan, I. Abunadi, M.E. JadHAV, S.M. Fati, Score and correlation coefficient-based feature selection for predicting heart failure diagnosis by using machine learning algorithms, *Comput. Math. Methods Med.* 2021 (Dec. 2021) 1–16, <https://doi.org/10.1155/2021/8500314>.
- [40] M. Diwakar, A. Tripathi, K. Joshi, M. Memoria, P. Singh, N. Kumar, Latest trends on heart disease prediction using machine learning and image fusion, *Mater. Today Proc.* 37 (2021) 3213–3218, <https://doi.org/10.1016/j.matpr.2020.09.078>.
- [41] R.J.P. Princy, S. Parthasarathy, P.S.H. Jose, A.R. Lakshminarayanan, S. Jeganathan, Prediction of cardiac disease using supervised machine learning algorithms, in: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 570–575, <https://doi.org/10.1109/ICICCS48265.2020.9121169>.
- [42] V. Shorewala, Early detection of coronary heart disease using ensemble techniques, *Inform. Med. Unlocked*. 26 (2021) 100655, <https://doi.org/10.1016/j.imu.2021.100655>.
- [43] F. Tasnim, S.U. Habiba, A comparative study on heart disease prediction using data mining techniques and feature selection, in: 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2021, pp. 338–341, <https://doi.org/10.1109/ICREST51555.2021.9331158>.
- [44] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, P. Singh, Prediction of heart disease using a combination of machine learning and deep learning, *Comput. Intell. Neurosci.* 2021 (Jul. 2021) 1–11, <https://doi.org/10.1155/2021/8387680>.
- [45] P.E. Rubini, C.A. Subasini, A.V. Katharine, V. Kumaresan, S.G. Kumar, T.M. Nithya, A cardiovascular disease prediction using machine learning algorithms, *Ann. Rom. Soc. Cell. Biol.* (2021) 904–912.
- [46] T. Rizinde, I. Ngaruye, N.D. Cahill, Comparing machine learning classifiers for predicting hospital readmission of heart failure patients in Rwanda, *J. Pers. Med.* 13 (9) (Sep. 2023) 1393, <https://doi.org/10.3390/jpm13091393>.
- [47] C. Boukhatem, H.Y. Youssef, A.B. Nassif, Heart disease prediction using machine learning, in: 2022 Advances in Science and Engineering Technology International Conferences (ASET), IEEE, 2022, pp. 1–6.
- [48] C.M. Bhatt, P. Patel, T. Ghelia, P.L. Mazzeo, Effective heart disease prediction using machine learning techniques, *Algorithms* 16 (2) (Feb. 2023) 88, <https://doi.org/10.3390/a16020088>.
- [49] A. Khan, M. Qureshi, M. Daniyal, K. Tawiah, A novel study on machine learning algorithm-based cardiovascular disease prediction, *Health Soc. Care Commun.* 2023 (Feb. 2023) 1–10, <https://doi.org/10.1155/2023/1406060>.
- [50] M. Kumar, A. Rai, Surbhit, N. Kumar, Autonomic edge cloud assisted framework for heart disease prediction using RF-LRG algorithm, *Multimed. Tools Appl.* 83 (2) (Jan. 2024) 5929–5953, <https://doi.org/10.1007/s11042-023-15736-9>.
- [51] N. Chandrasekhar, S. Peddakrishna, Enhancing heart disease prediction accuracy through machine learning techniques and optimization, *Processes* 11 (4) (Apr. 2023) 1210, <https://doi.org/10.3390/pr11041210>.
- [52] G. Saranya, A. Pravin, A novel feature selection approach with integrated feature sensitivity and feature correlation for improved prediction of heart disease, *J. Ambient. Intell. Humaniz. Comput.* 14 (9) (Sep. 2023) 12005–12019, <https://doi.org/10.1007/s12652-022-03750-y>.
- [53] M.C. Das, et al., A comparative study of machine learning approaches for heart stroke prediction, in: 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), 2023, pp. 1–6, <https://doi.org/10.1109/SmartNets58706.2023.10216049>.
- [54] G.A. Ansari, S.S. Bhat, M.D. Ansari, S. Ahmad, J. Nazeer, A.E.M. Eljaly, Performance evaluation of machine learning techniques (MLT) for heart disease prediction, *Comput. Math. Methods Med.* 2023 (May 2023) 1–10, <https://doi.org/10.1155/2023/8191261>.
- [55] L. Ashish, S.K. V, S. Yelgeti, WITHDRAWN: ischemic heart disease detection using support vector machine and extreme gradient boosting method, *Mater. Today Proc.* (Feb. 2021), <https://doi.org/10.1016/j.matpr.2021.01.715>.
- [56] V. Jain, M. Agrawal, Heart failure prediction using XGB classifier, logistic regression and support vector classifier, in: 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), 2023, pp. 1–5, <https://doi.org/10.1109/InCACCT57535.2023.10141752>.
- [57] J. Meehan, et al., Precision medicine and the role of biomarkers of radiotherapy response in breast cancer, *Front. Oncol.* 10 (2020) 628.
- [58] D. Rahmat, A.A. Putra, Hamrin, A.W. Setiawan, Heart disease prediction using K-nearest neighbor, in: 2021 International Conference on Electrical Engineering and Informatics (ICEEI), 2021, pp. 1–6, <https://doi.org/10.1109/ICEEI52609.2021.9611110>.
- [59] D.-J. Choi, J.J. Park, T. Ali, S. Lee, Artificial intelligence for the diagnosis of heart failure, *NPJ Digit. Med.* 3 (1) (2020) 54, <https://doi.org/10.1038/s41746-020-0261-3>.
- [60] O. Alsayyed, et al., Giant armadillo optimization: a new bio-inspired metaheuristic algorithm for solving optimization problems, *Biomimetics* 8 (8) (2023) 619.
- [61] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: a new method for stochastic optimization, *Futur. Gener. Comput. Syst.* 111 (Oct. 2020) 300–323, <https://doi.org/10.1016/j.future.2020.03.055>.
- [62] H. Yapıcı, N. Çetinkaya, A new meta-heuristic optimizer: pathfinder algorithm, *Appl. Soft Comput.* 78 (2019) 545–568, <https://doi.org/10.1016/j.asoc.2019.03.012>.
- [63] Q. Wu, A novel optimization algorithm: the forest algorithm, in: 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications, 2014, pp. 59–63, <https://doi.org/10.1109/ISDEA.2014.22>.
- [64] L.J. Webb, A physiognomic classification of Australian rain forests, *J. Ecol.* (1959) 551–570.
- [65] H. Gitay, I.R. Noble, J.H. Connell, Deriving functional types for rain-forest trees, *J. Veg. Sci.* 10 (5) (1999) 641–650.
- [66] C.J. Oliver, Rain forest classification based on SAR texture, *IEEE Trans. Geosci. Remote Sens.* 38 (2) (2000) 1095–1104.
- [67] M.L. Clark, D.A. Roberts, D.B. Clark, Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales, *Remote Sens. Environ.* 96 (3–4) (2005) 375–398.
- [68] S.E. Sesnie, B. Finegan, P.E. Gessler, S. Thessler, Z. Ramos Bendana, A.M.S. Smith, The multispectral separability of Costa Rican rainforest types with support vector machines and random Forest decision trees, *Int. J. Remote Sens.* 31 (11) (2010) 2885–2909.
- [69] S. Thessler, S. Sesnie, Z.S.R. Bendaña, K. Ruokolainen, E. Tomppo, B. Finegan, Using k-nn and discriminant analyses to classify rain forest types in a Landsat TM image over northern Costa Rica, *Remote Sens. Environ.* 112 (5) (2008) 2485–2494.
- [70] D.W. Hosmer Jr., S. Lemeshow, R.X. Sturdivant, *Applied Logistic Regression* vol. 398, John Wiley & Sons, 2013.
- [71] K. Özdamar, Tablolular Oluşturma, *Güvenirlik ve Soru Analizi Paket Programları İstatistiksel Veri Analizi-1*, Kaan Kitabevi, Eskişehir, 2004.
- [72] D.W. Hosmer Jr., S. Lemeshow, R.X. Sturdivant, *Applied Logistic Regression* vol. 398, John Wiley & Sons, 2013.
- [73] K. Özdamar, Tablolular Oluşturma, *Güvenirlik ve Soru Analizi Paket Programları İstatistiksel Veri Analizi-1*, Kaan Kitabevi, Eskişehir, 2004.
- [74] H.A. Abu Alfeilat, et al., Effects of distance measure choice on k-nearest neighbor classifier performance: a review, *Big Data* 7 (4) (2019) 221–248.
- [75] N. Suguna, K. Thanushkodi, An improved k-nearest neighbor classification using genetic algorithm, *Int. J. Comp. Sci. Issues* 7 (2) (2010) 18–21.
- [76] Y. Wu, K. Ianakiev, V. Govindaraju, Improved k-nearest neighbor classification, *Pattern Recogn.* 35 (10) (2002) 2311–2318.
- [77] Y. Jarraya, T. Madi, M. Debbaï, A survey and a layered taxonomy of software-defined networking, *IEEE Commun. Surv. Tutor.* 16 (4) (2014) 1955–1980.
- [78] M.J. Aitkenhead, A co-evolving decision tree classification method, *Expert Syst. Appl.* 34 (1) (2008) 18–25.
- [79] B. Charbuty, A. Abdulazeez, Classification based on decision tree algorithm for machine learning, *J. Appl. Sci. Technol. Trends* 2 (01) (2021) 20–28.
- [80] M.A. Friedl, C.E. Brodley, Decision tree classification of land cover from remotely sensed data, *Remote Sens. Environ.* 61 (3) (1997) 399–409.
- [81] A. Priyam, G.R. Abhijeeta, A. Rathhee, S. Srivastava, Comparative analysis of decision tree classification algorithms, *Int. J. Curr. Eng. Technol.* 3 (2) (2013) 334–337.
- [82] Y.-Y. Song, L.U. Ying, Decision tree methods: applications for classification and prediction, *Shanghai Arch. Psychiatry* 27 (2) (2015) 130.
- [83] Q. Wu, A novel optimization algorithm: the forest algorithm, in: 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications, 2014, pp. 59–63, <https://doi.org/10.1109/ISDEA.2014.22>.
- [84] M. Ghaemi, M.-R. Feizi-Derakhshi, Forest optimization algorithm, *Expert Syst. Appl.* 41 (15) (2014) 6676–6687.
- [85] H. Yapıcı, N. Çetinkaya, A new meta-heuristic optimizer: pathfinder algorithm, *Appl. Soft Comput.* 78 (May 2019) 545–568, <https://doi.org/10.1016/j.asoc.2019.03.012>.

- [86] A. Singh, R. Kumar, Heart disease prediction using machine learning algorithms, in: 2020 International Conference on Electrical and Electronics Engineering (ICE3), IEEE, 2020, pp. 452–457.
- [87] H. Jindal, S. Agrawal, R. Khera, R. Jain, P. Nagrath, Heart disease prediction using machine learning algorithms, in: IOP Conference Series: Materials Science and Engineering, IOP Publishing, 2021, p. 12072.
- [88] A. Garg, B. Sharma, R. Khan, Heart disease prediction using machine learning techniques, in: IOP Conference Series: Materials Science and Engineering, IOP Publishing, 2021, p. 12046.