Heart Disease Prediction Using Ensemble Methods, Genetic Algorithms, and Data Augmentation: A Preliminary Study

Deepali Yewale [1*], Swati Patil [2], Archana Rajesh Date [3], Aziz Nanthaamornphong [4]
[1] AISSMS Institute of Information Technology; Pune 411001, India
[2] Vishwakarma Institute of Technology; Pune, India
[3] HSBPVTS Faculty of Engineering; Ahmednagar, India
[4] College of Computing, Prince of Songkla University, Phuket Campus, Phuket 83120, Thailand
Email: [1] deepali_yewale@yahoo.co.in, [2] swati.patil@viit.ac.in, [3] archanadate@gmail.com, [4] aziz.n@phuket.psu.ac.th,
*Corresponding Author

***Abstract*—Statistically speaking, heart disease (HD) accounted for 1 in 5 fatalities in 2022, demanding affordable and accurate diagnosis. Traditional methods of prediction are accurate but expensive, creating a demand for sophisticated and efficient technologies. One of the most popular methods that researchers employ to forecast diseases is machine learning (ML). The goal of this effort is to improve HD prognosis accuracy through the use of ensemble approaches, specifically Random Forest (RF), XGBoost, Voting, and Stacking methods, which improve prediction accuracy by combining multiple models to capture complex patterns. Genetic algorithms (GA) are used to prioritize features. Incorporating data balancing, outlier removal techniques, and data augmentation, creates a model that delivers performance comparable to state-of-the-art research. Methods like random oversampling address data imbalance, while an isolation forest is employed to identify anomalies. To increase the dataset size and improve model performance, random noise is added after anomaly removal. Performed the cross-validation and robustness checks to assess the model's performance on both augmented and non-augmented datasets, ensuring that the inclusion of random noise did not excessively affect generalizability or result in overfitting. The proposed model's effectiveness is evaluated using various performance metrics. Achieving 99.36% accuracy, 98% sensitivity, 100% specificity, 100% PPV, 97% NPV, 0.99 F-score, and an AUC of 1, the methodology shows great promise as a cost-effective, accurate, and highly efficient diagnostic tool for heart disease. The model's short training time and high performance suggest its potential for practical implementation in clinical settings, offering a reliable and affordable solution for early heart disease detection.**

***Keywords*—Heart Disease; Ensemble Classifier; Genetic Algorithm; Data Balancing; Outlier Removal; Random Noise.**

I. INTRODUCTION

The WHO reports that cardiovascular diseases (CVD), which include coronary artery disease, heart failure, and arrhythmias claim millions of lives every year [1]. Early detection and prompt treatment are crucial in reducing the effects of HD, underscoring the importance of precise and reliable diagnostic approaches. Conventional diagnostic techniques, although effective, often rely on costly and resource-intensive tools such as invasive procedures or specialized imaging [2]. As healthcare systems worldwide strive for more affordable solutions, there is a growing demand for innovative technologies that offer precision without the associated high costs [3].

With the continuous evolution of medical systems and the development of novel treatments, it is increasingly difficult for healthcare professionals to keep pace with advancements in clinical practice. Providing effective care demands a thorough grasp of diagnostic guidelines, patient history, and an integrated approach to various therapeutic strategies. Yet, clinical decision-making is frequently shaped by intuition and past experiences, which can sometimes result in errors [4]. Consequently, computer-aided diagnostic systems have become valuable tools in assisting healthcare providers in reaching well-informed conclusions. The integration of ML techniques with medical expertise has garnered significant interest, as it can enhance diagnostic accuracy, prediction, and treatment. Research has demonstrated that machine learning algorithms can surpass even highly skilled physicians in diagnostic accuracy [5].

In the past few years, ML has become a highly regarded area for enhancing diagnostic precision in various medical applications, including HD prediction. ML algorithms are capable of analysing large quantities of patient information, identifying intricate patterns, and providing forecasts that support clinical decision-making [6]. Despite these advancements, challenges remain in terms of data quality, imbalance, and computational efficiency, all of which can affect model performance. Addressing these issues is essential to achieving the desired balance between accuracy, speed, and accessibility in real-world clinical environments [7].

The augmented datasets can help generalize models better, especially when dealing with small datasets. If the available dataset is limited, adding random noise to the data is a simple yet effective way to generate slightly different versions of the original data. This simulates new data points, effectively enlarging the dataset and making the model more robust [8]. GA is a powerful tool for feature optimization in machine learning, offering robust search capabilities across large feature spaces, especially in complex datasets. By iteratively selecting and refining feature subsets, GA helps in improving model performance, reducing overfitting, and simplifying models [9][10].

This research explores the development of a novel ensemble ML model, optimized by GA for feature selection (FS), to improve the prognosis of HD. The resulting models from genetic algorithms and augmented datasets can be highly intricate, posing challenges for clinicians who rely on transparent and explainable systems for decision-making. Ensuring model interpretability is crucial for gaining the trust of healthcare professionals and ensuring patient safety. The model integrates several advanced techniques, such as data balancing and outlier detection, to ensure robust performance on diverse datasets. By employing strategies like random oversampling to manage imbalanced data and using isolation forests for outlier detection, the study aims to overcome common limitations in current machine learning approaches. Furthermore, by adding random noise to double the dataset, the study seeks to improve model accuracy while maintaining computational efficiency. The addition of random noise during data augmentation can introduce artificial variability, which may not accurately reflect real-world clinical scenarios and could lead to overfitting. To mitigate this, we carefully monitored model performance across multiple validation sets to ensure that the augmentation process did not negatively impact the model's ability to generalize. The aim of this study is to develop a diagnostic instrument that not only matches cutting-edge performance but also offers an affordable, efficient solution for heart disease prediction in clinical settings. Here’s a bullet-point summary of the study's main contributions:

- Data Standardization: The research incorporates multiple data pre-processing techniques, including standardization, data balancing, and anomaly removal, to standardize the dataset.
- Feature Optimization: It utilizes a genetic algorithm to identify and prioritize the most valuable features, further enhancing diagnostic accuracy.
- Enhancement of Diagnosis: The study seeks to enhance the precision in diagnosing heart disease by employing an ensemble classifier in conjunction with data augmentation techniques.
- Comparative Analysis: The suggested approach is evaluated against current methods to highlight its superiority and performance in diagnostic accuracy.

The structure of the paper is as follows: An overview of pertinent material is provided in Section 2. Section 3 outlines the suggested approach, and Section 4 presents the results of the experiment. The conclusions are finally summed up in Section 5.

II. LITERATURE REVIEW

To investigate HD prediction, a range of data mining techniques have been employed, incorporating data augmentation and feature engineering strategies. This section provides a review and analysis of several key studies in this area.

Many researchers explored various ML methods, including base classifiers, ensemble classifiers, and hybrid methods for heart disease prognosis [11]-[14]. In the study [15], the researcher applied ensemble modelling techniques such as bagging, boosting, and stacking, comparing their performance against individual base classifiers. A comparative analysis was conducted to assess the effectiveness of ensemble methods in improving the prediction accuracy of coronary heart disease. Bagged models demonstrated higher accuracy compared to their traditional counterparts, while boosted models achieved an improvement in average accuracy. The stacked model, however, outperformed both, delivering the highest accuracy among all techniques. Additionally, the models’ performance was assessed through data analysis techniques and K-Fold cross-validation to ensure robustness.

Authors in the research paper [16], [17], suggested the method to effectively predict HD by developing a homogeneous ensemble model, specifically leveraging an accuracy-based weighted aging classifier. This approach obtained a classification accuracy of 93%, demonstrating its efficacy in predicting HD risk. The suggested method consistently outperformed other ML algorithms and recent methodologies, showcasing its superior predictive capability. This work sets a benchmark for using ensemble techniques in predictive healthcare systems.

Similarly, the study in [17], combined the effectiveness of ensemble learning, optimized through Bayesian hyperparameter tuning, with explainability through SHAP (Shapley Additive Explanations). The research evaluated three popular ensemble algorithms: AdaBoost, Random Forest, and XGBoost. XGBoost emerged as the best-performing algorithm, owing to its robust handling of structured datasets and computational efficiency. The combination of high specificity and sensitivity highlights the model’s reliability in accurately distinguishing between patients with and without HD. This study highlights how ensemble learning and examinability methods can be combined to achieve state-of-the-art disease prediction performance while maintaining model transparency.

CVDs account for a significant portion of global deaths, and their prevalence has escalated during the COVID-19 pandemic, due to the compounded effects of socio-economic challenges and lifestyle changes. The necessity for precise and timely diagnosis has become paramount in reducing the burden of heart disease. The research in [18], focused on using ensemble ML methods to predict HD, with FS, emphasizing the critical role in enhancing model performance. The results demonstrated that ensemble models, when optimized with selected features, achieved high accuracy with fewer input variables.

Due to the large volume of patient data, accurately predicting heart diseases remains challenging. Individual classification algorithms have been found insufficient for creating reliable technique for HD prediction. To address this, ensemble learning approaches (ELA) have been applied in [19], combining multiple algorithms to enhance predictive accuracy. The study utilized three datasets with diverse patient data for evaluating model generalizability.

The research underscores the importance of ensemble methods in creating robust and accurate predictive models for HD. This suggested approach demonstrated highest precision when evaluated using the Z-Alizadeh Sani dataset, outperforming its performance on the other datasets.

Researchers in [20], investigated the efficacy of several ML methods and their combination on benchmark dataset in cardiovascular research to enhance prediction accuracy. GridSearchCV with five-fold cross-validation systematically explored parameter combinations to maximize model performance. LR and AdaBoost algorithms were combined into a soft voting ensemble classifier, resulted in higher accuracies, reaching 93.44% and 95%, outperforming previous approaches for the said two datasets. This research establishes a strong case for integrating machine learning algorithms into ensemble frameworks, showcasing their ability to handle diverse datasets and accomplish cutting-edge results.

Researchers in [21], emphasized on data pre-processing techniques, and the use of ensemble learning algorithms in HD management. To produce a complete dataset for analysis, the study combined many Kaggle datasets with comparable properties, and employed GridSearchCV to tune the hyperparameters systematically. By applying the Extra Tree Classifier, this approach achieved a high prediction accuracy of 98.15%. The findings emphasize the value of thorough pre-processing as well as the competency of ML models to enhance healthcare early detection.

In the study [22], researchers implemented a stacked ensemble classifier to improve the precision of disease prediction diagnosis, leveraging various ML approaches and FS methods. Chi-square was applied for feature optimization to identify the most statistically significant features, reducing dimensionality and improving computational performance. This research emphasizes the importance of integrating FS and advanced ensemble techniques to build robust models, offering valuable insights for practical healthcare applications.

In previous studies, ensemble learning systems were employed for HD risk prediction by exploring multiple machine learning models. The stack-based ensemble approach in [23] combined weak learners to predict early HD symptoms. A meta-classifier, specifically LR, was used to integrate the predictions from these models. SMOTE was applied to address disparity in class for UCI dataset. The method attained an AUC score of 0.922, and performance was further evaluated using confusion matrix and classification reports. This research illustrates the effectiveness of stack-based ensembles and preprocessing techniques like SMOTE in advancing predictive accuracy and fairness in HD risk assessment.

In the study [24], researchers implemented an ensemble ML approach for CVD detection, leveraging a voting mechanism to combine the strengths of multiple classifiers. Chi-square was applied to reduce the feature set, improving both accuracy and reducing computational complexity. The ensemble method showed superior performance, attaining a noteworthy accuracy of 92.11%, demonstrating improvement over any single classifiers. This research showcases the potential of ensemble learning in advancing diagnostic tools, emphasizing the synergy between FS and model combination to achieve superior performance.

In previous studies, various ML techniques have been applied to predict HD using clinical datasets. FS methods, including Pearson, PCA, Chi-2, and RFE, have been employed to identify the most significant clinical attributes, followed by an ensemble approach to further refine the feature set. A number of machine learning classifiers have had their performance assessed, including artificial neural networks, ensemble (bagging and boosting), and conventional classifiers. The results underscore the potential of boosting algorithms like XGBoost in clinical applications, providing a trustworthy and effective instrument for early diagnosis and management [25]. This study emphasizes the synergy between effective FS and ensemble learning in developing robust, high-performing models for healthcare management.

The researchers in [26], focused on evaluating the effect of FS techniques on the efficiency of various ML models for heart risk assessment.The study underscored that the effect of FS varies across algorithms, emphasizing the need to tailor FS techniques to specific models. This study emphasizes how important FS is for improving the functionality of specific models, while also demonstrating its potential drawbacks for others. The findings provide valuable insights for selecting appropriate FS methods based on the model and dataset characteristics. The study emphasizes the nuanced relationship between FS and model performance, advocating for a careful, model-specific approach in predictive modeling for HD.

The study [27] performed an experimental evaluation of HDAll things considered, this study underscores how crucial it is to integrate advanced ML strategies to address healthcare challenges prediction models, focusing on the interplay between FS techniques and classification algorithms. A total of ten FS methods, were assessed alongside six classification approaches for the UCI dataset. The results indicated that the backward FS technique produced the best-performing feature subset, yielding improved model accuracy and computational efficiency. This comprehensive analysis contributed to the understanding of data-driven methodologies in predicting HD.

The authors in [28], explored the difficulties associated with detecting anomalies in time-series data, which often suffers from issues like data imbalance, temporal dependence, and noise. They proposed a fault detection model that utilized data augmentation through the addition of Gaussian noise, set at a level of 0.002, to enhance the model's generalization performance. Their evaluation indicated high F1-scores and accuracy when time series data was processed, thereby providing a foundation for further research in anomaly detection.

The study referenced in [29], proposed a innovative methodology that combines reinforcement learning and data augmentation techniques to increase the prediction accuracy of heart illness. This approach tackled the inherent challenges in cardiac data, such as its complexity and variability, which often hinder the performance of conventional ML models. The combination of these two approaches resulted in a synergistic effect, improving the model's capacity to accurately forecast outcomes and capture intricate data linkagesAll things considered, this study underscores how crucial it is to integrate advanced ML strategies to address healthcare challenges and sets a benchmark for future research in cardiac disease prediction.

Our approach stands out by emphasizing both high accuracy and computational efficiency in predicting heart disease using an ensemble classifier with feature optimization. The dataset undergoes data balancing and outlier removal to ensure better quality, and data augmentation techniques are applied by introducing random noise during training to enhance model robustness and performance in binary classification.

III. Materials and Methods

A. *Dataset*

For the experimentation in this research, the dataset is sourced from the UCI Machine Learning Repository [30]. It includes 303 entries and initially contains 76 features; however, only fourteen of these biological characteristic are commonly used in prediction of heart risk [31]. Of these, thirteen represent input features related to various health parameters, regarded as risk factors for HD. With values of 0 or 1, the binary target output indicates if heart risk is present or not. Table I provides a description of the dataset's biological Characteristics.

TABLE I. Biological Characteristics Description of the Dataset

Feature	Data Type	Description
age	Numerical	Age in years
sex	Categorical	Male: 0; Female: 1
cp	Categorical	Type of chest pain
trestbps	Numerical	Resting blood pressure
chol	Numerical	Serum cholesterol
fbs	Categorical	Fasting blood sugar
restecg	Categorical	Resting ECG result
thalach	Numerical	Maximum value of heart rate
exang	Categorical	Exercise induced angina
oldpeak	Numerical	ST depression induced by exercise
slope	Categorical	The slope of the peak ST segment
ca	Numerical	Number of major vessels colored by fluroscopy
thal	Categorical	Thalasemia
target	Categorical	Diagnosis of heart disease

B. *Data Pre-processing*

To guarantee the integrity of the dataset and its compliance with the ML model, the data pre-processing stage is essential. Initially, the process involves identifying and handling missing values, which can undermine the statistical reliability and accuracy of the model's predictions [32]. We identified no missing values in the dataset, eliminating the need for imputation or exclusion methods. However, a further inspection identified disparities in the magnitude of certain attribute values. The chosen method, standardization, transforms the values of each attribute to have a mean of 0 and a standard deviation of 1 [33]. This ensures that all attributes contribute equally to the model during training. Each feature's data point's mean is deducted during standardization, and the result is divided by the standard deviation. The standardization process is mathematically represented using (1) to (3),

$$x' = \frac{x-\mu}{\sigma} \tag{1}$$

Where, x is the data point in the feature, μ the mean, and σ is the standard deviation.

Here, mean is computed as:

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2}$$

In the context of a dataset, N is the total number of instances being scaled, and x_i represents each individual data point in the feature.

The standard scalar is calculated as:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2} \tag{3}$$

C. *Random Oversampling*

As seen in Fig. 1, the Cleveland dataset has 165 individuals diagnosed with HD, whereas 138 individuals do not have the disease. For instance, in medical diagnostics, the minority class might represent a rare but critical disease that requires detection. Understanding and addressing imbalance is essential because standard machine learning algorithms can become biased in favor of the dominant class, resulting in subpar minority class prediction performance [34]. So, the dataset is balanced by the use of random oversampling. In order to balance the minority class with the majority class, RandomOverSampler duplicates samples from the minority class [35]. Fig. 2 represents the class distribution after resampling. Total Number of instances after random oversampling is 330.

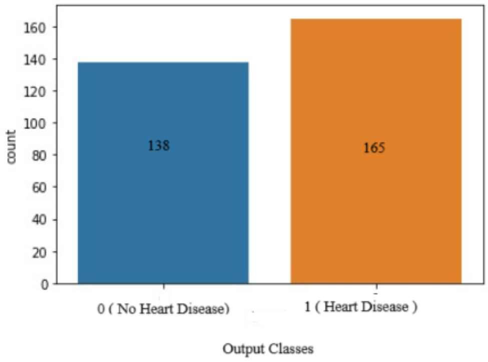


Fig. 1. Target class distribution of cleveland dataset

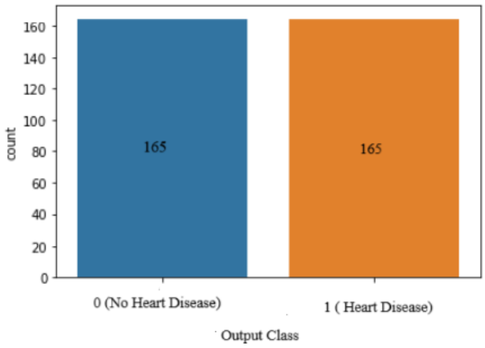


Fig. 2. Target class distribution of cleveland dataset

D. Isoaltion Forest

Isolation Forest is a specialized technique for detecting anomalies in data [36]. It identifies anomalies by isolating observations, where "isolating" means separating a data point from the rest. Isolation Forest uses decision trees to partition data [37]. The algorithm builds these trees through a process of randomization, where it:

- Randomly selects a feature from the dataset.
- Randomly chooses a split value within the range defined by the minimum and maximum values of the selected feature.

This process is repeated recursively to construct individual decision trees The algorithm builds an ensemble of these random partitioning trees and averages the "isolation path length" across the trees [38]. Short paths indicate anomalies, while longer paths indicate normal points.

When the Isolation Forest model (iForest) is fitted to a dataset, it creates 50 trees (based on n_estimators=50), each randomly partitioning the data. After training, each data point is scored based on how many splits are needed to isolate it across all trees. Points with a shorter average path length are classified as outliers, while those with longer path lengths are classified as normal [39]. With contamination=0.05, the model will attempt to label 5% of the data as anomalies.

The designed code fits the Isolation Forest model on a resampled dataset and then makes predictions, where it assigns -1 to anomalies and 1 to normal points.

Cleaned dataset with 313 instances after removal of outliers is further processed for data augmentation.

E. Data Augmentation

To further expand the diversity and size of the dataset, random noise is added to the instances, effectively doubling the data available for training [40]. This augmentation technique introduces variability in the input features, enhancing the model’s robustness and generalization capabilities [41]. As the available dataset is limited, adding random noise to the data is a simple yet effective way to generate slightly different versions of the original data. This simulates new data points, effectively enlarging the dataset and making the model more robust.

np.random.normal() function generates random numbers from a normal (Gaussian) distribution. After generating the random values, they are added element-wise to the old_data. This effectively adds a small amount of noise (with a standard deviation of 0.1 and a mean of 0) to the existing old_data. This level of noise is chosen to introduce a moderate amount of variability without significantly altering the characteristics of the original data. The shape of the random noise matches the shape of X, and the element-wise addition adjusts each value in old_data. Some models exhibit sensitivity to small fluctuation in the input data. Adding noise makes the training process less sensitive to small fluctuations, promoting stability in predictions [42]. Cross validation is performed to ensure the integrity in the original data pattern. Compared the model performance with and without noise augmentation using a validation set, as well as implemented robust validation techniques to evaluate generalizability across different dataset size to ensure that the augmentation process did not negatively impact the model's ability to generalize. Total numbers of available instances after data augmentation is 626.

F. Feature Optimization using Genetic Algorithm

Genetic Algorithms (GA) are popular for feature optimization in machine learning because of their ability to efficiently search large, complex spaces [43]. They prioritize the most fit individuals for reproduction in order to produce the next generation, simulating the process of natural selection. this concept is applied to optimize the selection of features that improve model performance [44].

Genetic algorithm in Feature optimization [45]:

- Initialization: A population of chromosomes (each representing a feature subset) is initialized. Each chromosome is represented as a binary vector, where a feature is considered included if its value is 1, and excluded if its value is 0.
- Fitness Function: By training an ML model with the relevant subset of characteristics and assessing its performance, the fitness of each chromosome is assessed.
- Selection: Chromosomes with higher fitness scores are selected to produce offspring for the next generation. Techniques like roulette wheel selection or tournament selection are often used.
- Crossover: To create offspring’s, pairs of chromosomes are joined, usually by switching some of their feature subsets (e.g., via single-point or multi-point crossover). This mimics biological reproduction and encourages the exchange of useful feature combinations.
- Mutation: Random changes are introduced to the offspring by flipping a small number of bits in their chromosome. Random modifications are applied to certain individuals, promoting diversity and preventing the algorithm from becoming stuck in local optima.
- Termination: The process repeats for several generations until a stopping criterion is reached, such as a maximum number of generations or no improvement in fitness over time.
- Result: The best-performing chromosome represents the optimized feature subset, which can be used to train the final ML model.

In HD prediction, a genetic algorithm could optimize the feature set by selecting relevant clinical parameters (like cholesterol, age, blood pressure) that maximize model accuracy while excluding redundant or irrelevant features.

The stopping criteria in a genetic algorithm dictate when the algorithm should halt its execution. This is usually based on factors such as the maximum number of generations, achieving a satisfactory fitness level, or convergence of the population. When setting the parameters like stopping criteria, population initialization, and mutation/crossover rates, it's important to consider both the size and complexity of the dataset. In this case, using a population of 80 (`size=80`) and setting the mutation rate to 0.20 is intended to maintain a balance between exploration and exploitation, ensuring the algorithm explores a diverse set of potential solutions without excessive randomness. Furthermore, as the algorithm is run for only one generation (`n_gen=1`), these parameters are likely selected to quickly evaluate the impact of the GA with minimal computational cost.

In this example, the Genetic Algorithm runs for one generation with a population of 80 individuals, 64 parents selected for crossover, a mutation rate of 20%, and uses the provided training and testing datasets. The choice of parameters ensures a balance between diversity in the population (via random initialization and mutation) and convergence toward better solutions (via crossover). Evaluated the model performance using both training and validation sets, ensuring that the final feature set are not overfitted to the training data.

G. *Model Training with Ensemble Techniques*

The optimized feature set is then used to train an ensemble model, combining multiple base learners to improve predictive accuracy and reduce variance [46]. The fundamental idea behind ensemble learning is to harness the collective strength of various individual models, thereby creating a predictive system that is more accurate and reliable [47]. In this research work, several ensemble techniques were employed, including bagging with RF, ETC, boosting with XGBoost and AdaBoost, voting using both hard and soft voting methods, and stacking by combining RF with SVM.

The ensemble technique includes [48]:

- Bagging: Models are trained on different subsets of data with replacement, enhancing model stability and reducing variance.
- Boosting: Sequentially corrects the errors of previous models, emphasizing misclassified instances, thereby increasing overall accuracy.
- Voting: Each base model to "vote" on the outcome for a given input, with the ensemble model deciding based on these votes.
- Stacking: Combines predictions from multiple base models, using a meta-learner to enhance performance by leveraging the strengths of different algorithms.

1) *Bagging- Random Forest*

Random Forest, a robust ensemble learning method with the Bagging approach, constructs numerous decision trees as shown in Fig. 3. Each decision tree in the ensemble is trained independently on its respective subset, which is formed through bootstrapping. Each subset may include duplicate samples, ensuring diversity in the training data provided to individual trees. This randomization leads to a more generalized model by reducing correlation among trees. The bagging process is repeated multiple times, resulting in a collection of decision trees forming the forest. Once the independent decision trees are created, the random forest can perform a majority vote for classification tasks [49].

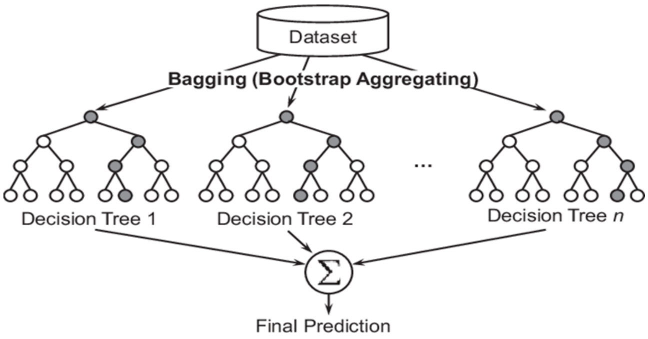


Fig. 3. Illustration of Random Forest [50]

2) *Bagging- Extra Tree Classifier*

It is an extension of Random Forest and shares some similarities with it. Using bootstrapping, Extra Trees creates several decision trees, each trained on a different subset of the original dataset as depicted in Fig. 4. Random Forest employs the optimal splitting to select the best feature from the random subset at each node while Extra Trees randomly selects features and thresholds for splitting without considering the optimal split.The final prediction is obtained by aggregating individual tree outputs through majority voting for classification tasks [51]. During training, the ensemble also evaluates performance using out-of-bag samples, and hyperparameter tuning can be used to optimize model performance.

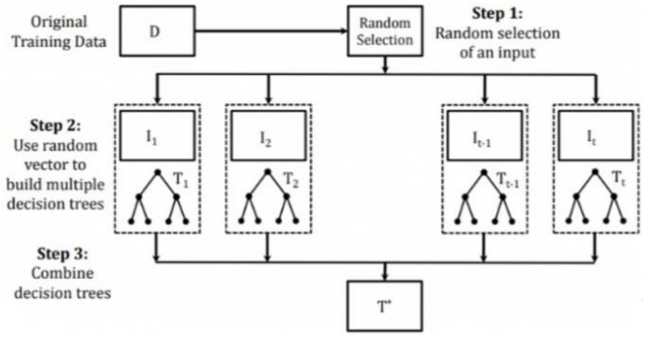

Original Training Data
D
Random Selection
Step 1: Random selection of an input
Step 2: Use random vector to build multiple decision trees
I_1
I_2
I_{t-1}
I_t
T_1
T_2
T_{t-1}
T_t
Step 3: Combine decision trees
T'

Fig. 4. Illustration of Extra Tree Classifier [52]

3) *Boosting- XGBoost*

Extreme Gradient Boosting, or XGBoost, is a well-known and highly regarded gradient boosting technique. A sequence of weak learners is trained in this manner, with each succeeding learner concentrating on fixing the errors of its predecessors [53] as represented in Fig. 5. Misclassified samples are given additional weight during training so that poor learners can improve their ability to recognize intricate patterns in the data. The outputs of all weak learners are combined to provide the final prediction, where the weights are determined by the individual performance of the learners. XGBoost employs a greedy algorithm to determine the best possible split. An essential aspect of XGBoost is that each new classifier considers the areas where the previous classifiers struggled to perform well [54].

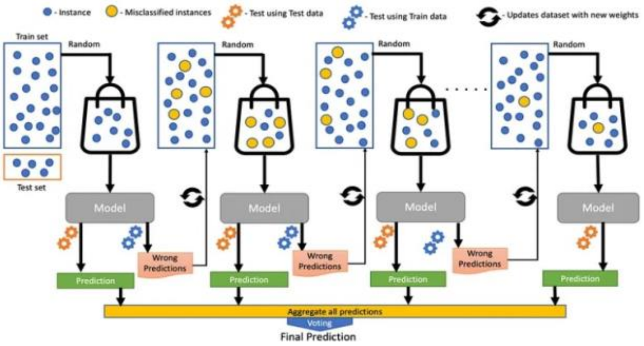


Fig. 5. Schematic representation of XGBoost [55]

4) *Boosting- AdaBoost*

AdaBoost is a potent machine learning technique that uses weighted linear combinations to combine several weak learners in order to produce a strong classifier. The weights of the training data instances are iteratively changed according to the classification accuracy of the data. Initially, all instances have equal weight. With each iteration, misclassified examples receive higher weights, while correctly classified ones receive lower weights. The base classifier is then applied to the updated data with fresh weights in each round [56]. All of the models created over the iterations are combined to create the final categorization model as illustrated in Fig. 6.

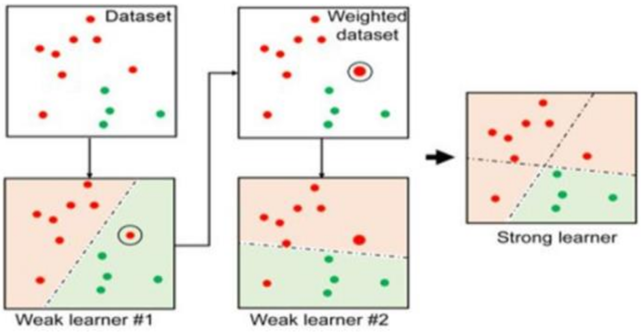
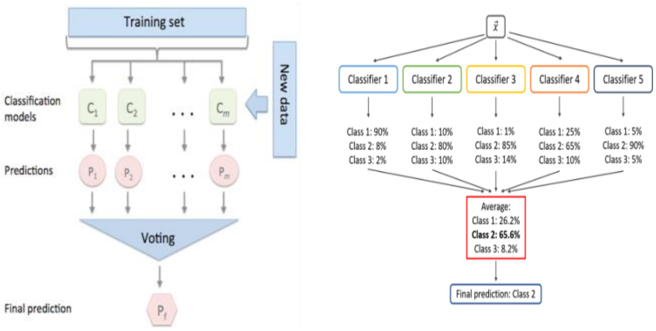


Fig. 6. Illustration of AdaBoost [57]

5) *Voting Classifier*

The Voting Classifier is a technique in ensemble learning that integrates several individual classifiers to generate predictions for a specific dataset. The initial step in constructing a Voting Classifier involves the careful selection of a group of base classifiers. These classifiers can be different machine learning algorithms or distinct instantiations of the same algorithm with varying hyperparameters. Once the base classifiers are chosen, they undergo training using the identical dataset and the same input features. After training the Voting Classifier, it can be used to predict outcomes on fresh and previously unseen data. The input data is passed through each base classifier, and their individual predictions are combined using either hard or soft voting strategy to generate the final prediction [58]. Hard voting uses a straightforward majority voting method, with the class with the most votes becoming the predicted result as shown in Fig. 7(a). Each classifier's class probabilities are averaged during soft voting, and the projected class is chosen based on the class with the highest average probability, as illustrated in Fig. 7(b). For hard voting, the ensemble of base models in the proposed work consists of LR, NB, DT, SVM, and KNN; for soft voting, it consists of LR, NB, DT, and KNN.



(a) (b)

Fig. 7. Illustration of (a) hard voting (b) soft voting [59]

6) *Stacking*

Stacking, also referred to as "Stacked Generalization," is an ensemble learning technique that enhances predictive performance by combining the outputs of multiple base models, also known as first-level models. This approach involves training several base models on the same training dataset and then using their predictions as input for a higher-level model, called a meta-model or second-level model, to generate the final prediction [60] as explained in Fig. 8. The core idea of stacking is to leverage the strengths of diverse base models to achieve superior predictive accuracy compared to using any single model. To ensure this diversity, the base models are either trained using different algorithms or by applying the same algorithm with distinct sets of hyperparameters. After training the base models on the training dataset, they are used to make predictions on the validation set. These predictions from the base models are then utilized as input features for the meta-model. The meta-model, trained on these features, learns to optimally combine the outputs of the base models to make the final prediction. Once trained, the meta-model is used to generate predictions on the test dataset [61]. In this research, RF is employed as a baseline model, while SVM is used as the meta-classifier to aggregate the predictions from the base models effectively.

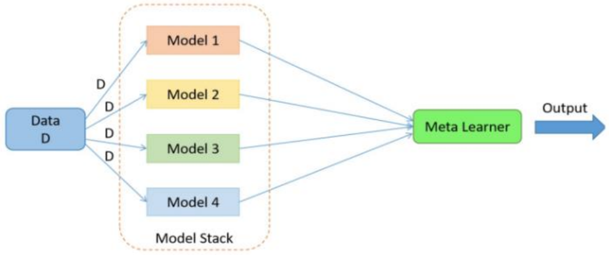


Fig. 8. Illustration of Stacking Approach [62]

H. *Proposed Prediciton Model*

The proposed research leverages ensemble learning techniques combined with various supporting algorithms for more precise heart disease detection. Given the medical disease prediction context, a large and complex dataset is involved. Machine learning models often face challenges like outlier detection and data imbalance. To address data imbalance, random oversampling is applied, while outliers are managed using the Isolation Forest algorithm. For feature optimization, a Genetic Algorithm (GA) is employed to prioritize the most relevant features. We also tested additional feature selection technique, such as RFE, but found that their inclusion did not significantly improve model performance over the GA alone. Data augmentation, in the form of random noise addition, is used to increase the dataset size while maintaining computational efficiency. The study employs ensemble classifiers, including bagging, boosting, voting, and stacking, to enhance predictive accuracy. The performance of these ensemble models is thoroughly compared. The ensemble learning framework combines multiple classifiers, including Random Forest (RF), XGBoost, Voting, and Stacking, with exploration of other ensemble techniques such as AdaBoost and Gradient Boosting. However, the performance of these alternative methods did not substantially outperform the existing approach, suggesting that the current ensemble setup offers a robust solution for heart disease prediction. Fig. 9 provides a visual representation of model's components, illustrating its structure and workflow. Subsequent sections delve into the study's methodology, covering dataset description, pre-processing methods, ensemble strategies aimed at boosting diagnostic accuracy, and feature optimization techniques to improve classifier performance.

The proposed methodology is outlined in detail in the following algorithm.

Algorithm: Proposed Approach

```
Step 1: Data Pre-processing
    1. Load Dataset: Import the heart disease dataset
    2. Data Cleaning:
        o Handle missing values by imputing or removing them based on the dataset's quality and the impact on feature importance.
    3. Data Balancing:
        o Apply Random Over-Sampling to handle imbalanced classes.
    4. Outlier Detection and Removal:
        o Use Isolation Forest to identify and eliminate outliers that may impact model performance.
Step 2: Data Augmentation
    1. Instance Doubling with Noise:
        o To enhance data diversity, add small random noise to existing instances and duplicate them to double the dataset size.
Step 3: Feature optimization Using GA
    1. Initialize Population:
        o Generate a population of feature subsets, with each subset representing a unique combination of features.
    2. Fitness Function:
        o  Establish a fitness function to assess feature subsets according to the model's sensitivity, specificity, and accuracy.
    3. Genetic Operators:
        o Apply Selection, Crossover, and Mutation to create new feature subsets for subsequent generations
        o Continue until convergence or maximum generations are reached.
    4. Select Optimal Features:
        o Select the feature subset that achieves the highest fitness score.
Step 4: Model Development and Ensemble Technique
    1. Split Data:
        o Divide the dataset into training and testing sets in a 75:25 ratio.
        o Apply 10-fold cross validation for validating the training set.
    2. Model Selection and Ensemble:
        o Develop an ensemble of machine learning models
        o Train each model on the train set utilizing the chosen features.
Step 5: Model Evaluation
    1. Prediction on Test Set:
        o Use the ensemble model to predict heart disease on the test set.
    2. Performance Metrics Calculation:
        o Assess the model's performance using various metrics for comparison.
Step 6: Result Interpretation
```

After training and validating the models across various evaluation metrics, the model achieving the highest accuracy is selected as the optimal one. The flowchart is designed to provide a better visual understanding of the methodology as shown in Fig. 10. This transparency is especially valuable in clinical environments, as it helps healthcare practitioners grasp the model’s decision-making process, fostering both trust and practical insights.

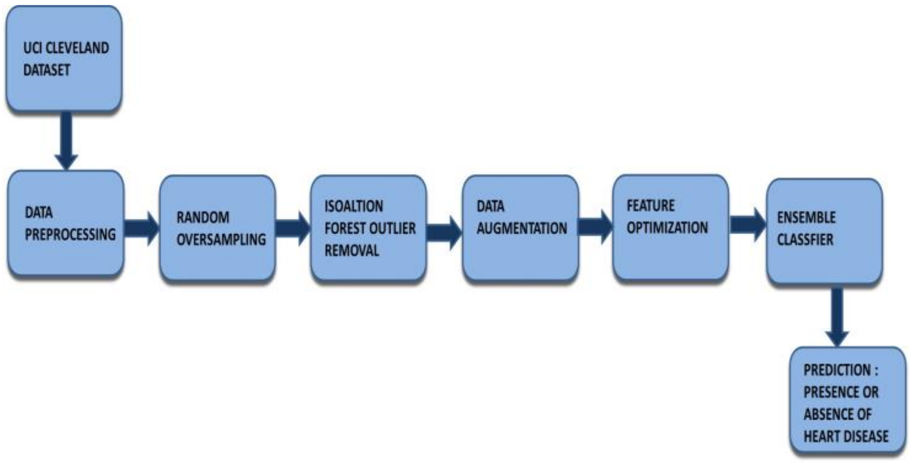


Fig. 9. Proposed Heart disease Prediction Model

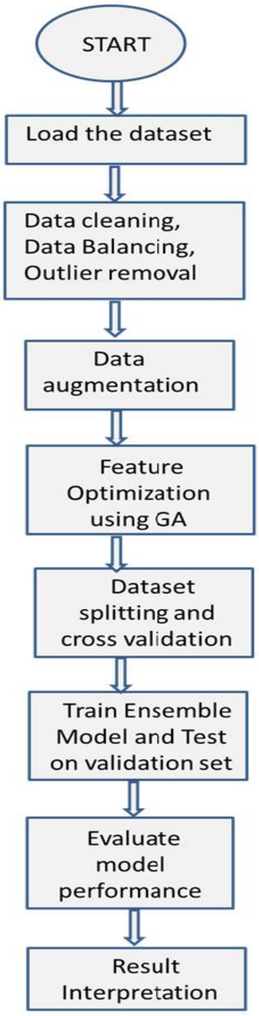

START
Load the dataset
Data cleaning, Data Balancing, Outlier removal
Data augmentation
Feature Optimization using GA
Dataset splitting and cross validation
Train Ensemble Model and Test on validation set
Evaluate model performance
Result Interpretation

Fig. 10. Flowchart representation of the proposed methodology

IV. Results and Discussion

Here, we examine the performance outcomes of proposed ensemble classifiers. Using the Cleveland dataset, we assessed these classifiers both before and after applying data augmentation and feature optimization with a genetic algorithm. The dataset was partitioned into train and test sets with a 75:25 split, ensuring that the test set was reserved exclusively for assessing the performance of the implemented ensemble classifiers. This separation helps mitigate overfitting risks associated with combining FS and ensemble methods. Python 3 and Jupyter notebook running on an i5- 9000H laptop with 8GB RAM are used to simulate the model.

The experiments involved two distinct methodologies. In the first approach, the process was carried out up to the outlier removal stage, after which the ensemble classifiers were trained individually and evaluated based on thirteen selected features.

The results for this approach 1 are shown in Table II. Among the ensemble classifiers, XGBoost, Soft Voting, and Stacking achieved outstanding performance, with an accuracy of 98.73%. We further assessed the classifiers using metrics such as sensitivity, specificity, PPV, NPV, and F1-score. Sensitivity reflects the classifier's ability to identify all positive cases, which is crucial for recognizing the majority of heart disease patients and reducing misdiagnoses. However, in medical situations, where a high false-positive rate might cause needless worry, expenses, and additional tests for individuals, specificity is crucial. In a heart disease prediction model, high specificity means that fewer healthy people are mistakenly labeled as having the disease. PPV indicates the classifier's accuracy in identifying true positives from predicted positives. In heart disease prediction, higher PPV means a lower false-positive rate, reducing unnecessary worry or medical testing for patients. High NPV allows negative test results to be trusted; for example, in a heart disease prediction model, a high NPV suggests that people predicted as disease-free likely do not have the disease, minimizing the need for additional testing. Finally, the F1-score is an excellent measure of classifier performance when recall and precision are balanced.

According to Table II, RF, ETC, and AdaBoost each had a 97.47% accuracy rate, 95% sensitivity, 100% specificity, 100% PPV, 95% NPV, and a 97% F1-score. The corresponding AUC values were, in order, 0.998, 0.982, and 0.995. Similarly, 98.73% accuracy was attained via XGBoost, hard voting, soft voting, and stacking, with 98% sensitivity, 100% specificity, 100% PPV, 97% NPV, and 99% F1-score. The corresponding AUC values for XGBoost, soft voting, and stacking were 0.995, 1, and 0.98. However, the stacking Ensemble classifier managed to acquire a better AUC of 1 with less computational time of 0.011 sec.

In the second approach, the dataset processed through outlier removal in the first approach underwent data augmentation, effectively doubling the number of instances. Following this, a genetic algorithm was applied for feature optimization. A genetic algorithm (GA) optimizes features in the Cleveland Heart Disease dataset by iteratively selecting and refining subsets of features that make up the majority of the model's predictive ability. The same ensemble classifiers used in the first approach were then trained on this newly optimized dataset and evaluated across various performance metrics. A GA runs through multiple generations to gradually improve the feature set. Each generation involves evaluating multiple feature subsets (chromosomes), and for each chromosome, the model is trained and tested to calculate a fitness score. This iterative process can significantly increase computational load.

TABLE II. PERFORMANCE EVALUATION FOR CLEVELAND DATASET USING APPROACH 1

Metric	RF	ETC	XGBoost	AdaBoost	Hard Voting	Soft Voting	Stacking
Accuracy	97.47	97.47	98.73	97.47	98.73	98.73	98.73
Sensitivity	95	95	98	95	98	98	98
Specificity	100	100	100	100	100	100	100
PPV	100	100	100	100	100	100	100
NPV	95	95	97	95	97	97	97
F1-Score	0.97	0.97	0.99	0.97	0.99	0.99	0.99
AUC	0.998	0.982	0.995	0.995	-	1	0.980
Computational Time in sec.	0.054	0.062	0.144	0.114	0.016	0.011	0.631

The performance results for this approach are presented in Table III. The findings demonstrate that feature optimization and data augmentation enhanced performance as compared to approach 1, but had increased the computational load.

Table III showed that all the optimized models managed to attain higher accuracy as compared to approach 1. In approach 2, RF has attained accuracy 99.36% with sensitivity 98%, specificity 100%, PPV 100%, NPV 97%, F1-score 99%, and AUC 0.998 with less computational time 0.071 sec. ETC has attained accuracy 98.09% with sensitivity 97%, specificity 100%, PPV 100%, NPV 96%, F1-score 99%, and AUC 0.997 with computational time 0.0937 sec. XGBoost has attained accuracy 99.36% with sensitivity 99%, specificity 99%, PPV 99%, NPV 99%, F1-score 99%, and AUC 0.998 with computational time 0.172 sec. AdaBoost has attained accuracy 98.73% with sensitivity 98%, specificity 100%, PPV 100%, NPV 97%, F1-score 99%, and AUC 0.992 with computational time 0.130 sec. Hard voting has attained accuracy 98.73% with sensitivity 98%, specificity 100%, PPV 100%, NPV 97%, F1-score 99%, with computational time 0.643 sec. Soft voting has attained accuracy 99.36% with sensitivity 98%, specificity 100%, PPV 100%, NPV 97%, F1-score 99%, and AUC 1 with computational time 0.652 sec. Stacking has attained accuracy 99.36% with sensitivity 97%, specificity 100%, PPV 100%, NPV 96%, F1-score 99%, and AUC 0.986 with computational time 1.096 sec.

The models trained with augmented data still perform well on the independent validation sets, making sure the augmentation didn't just memorably fit the training data. Implemented cross-validation (k-fold) to assess model performance with approach 1 and 2. Reexamined the AUC values, and after applying regularization techniques, we observed that the scores are now more consistent and realistic.

We have validated our models on the non_augmented Cleveland dataset in approach 1. The results on this dataset indicate that our models are generalizing well and are not overfitting to the augmented _Cleveland dataset in approach 2.

The graphical display of all the performance parameters of the suggested methodology, including all of the ensemble techniques used, is examined in Fig. 11. It shows that when applied to the Cleveland dataset with feature optimization via GA, all of the ensemble approaches perform exceptionally well.

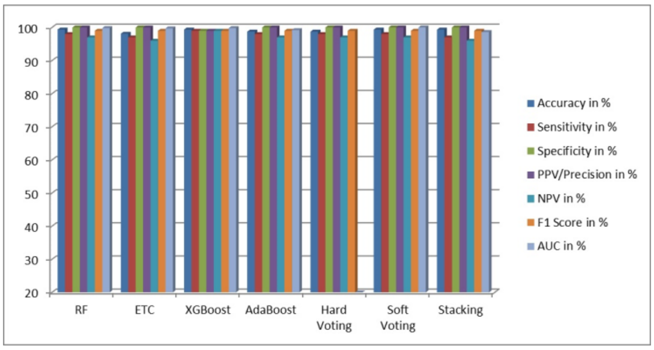
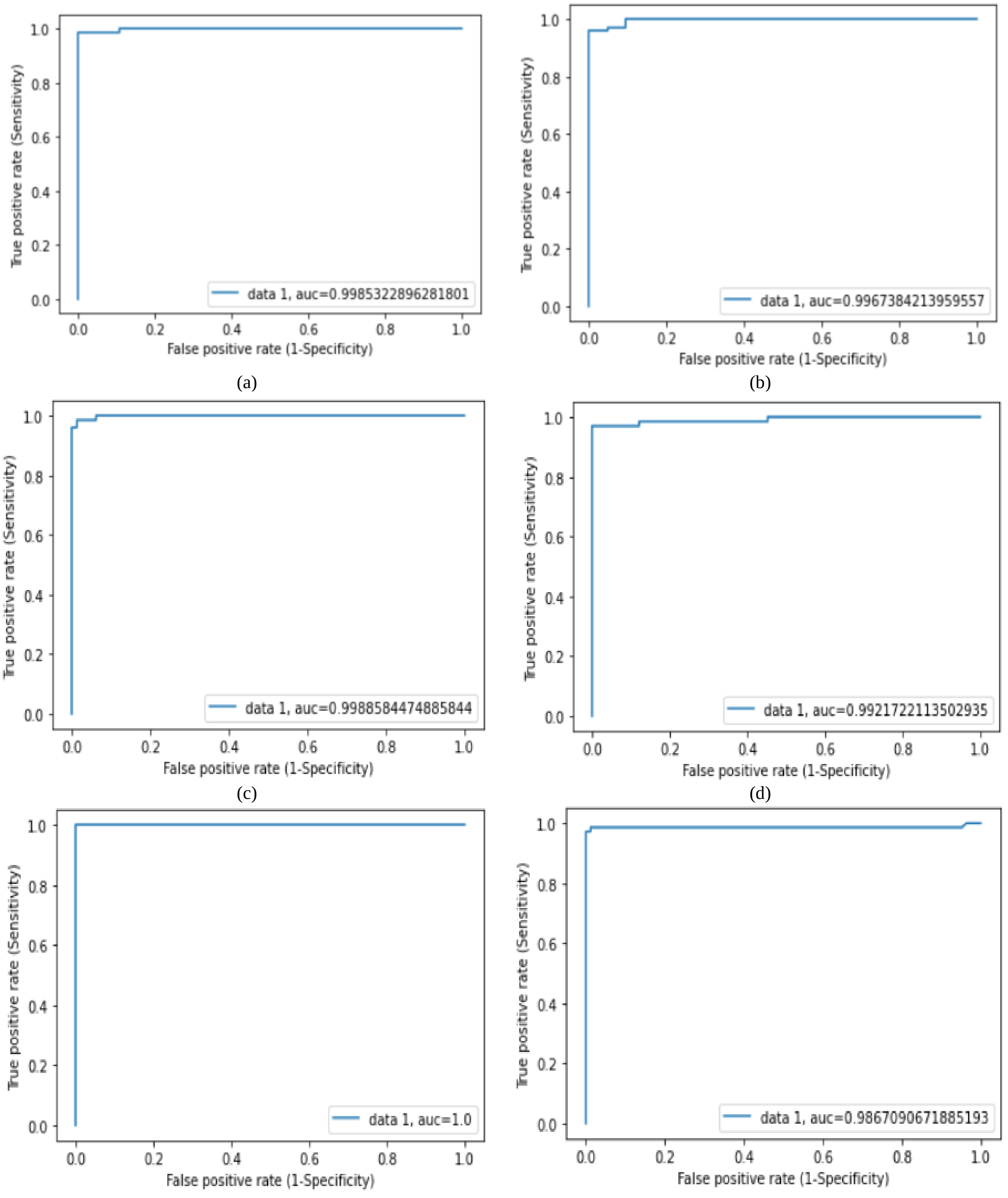


Fig. 11. Performance metrics of the suggested model

The proposed approach is further evaluated through ROC-AUC analysis. The ROC curve demonstrates the connection between the true positive rate and the false positive rate, effectively representing the balance between sensitivity and specificity. An AUC score between 0.9 and 1 is generally considered to indicate outstanding performance. Fig. 12 displays the ROC-AUC curves for different ensemble methods used in the experiments. For binary classification tasks, accuracy and AUC are crucial metrics. According to the ROC curve results, all methods achieved impressive AUC values, indicating excellent model performance.

TABLE III. PERFORMANCE EVALUATION FOR CLEVELAND DATASET USING APPROACH 2

Metric	RF	ETC	XGBoost	AdaBoost	Hard Voting	Soft Voting	Stacking
Accuracy	99.36	98.09	99.36	98.73	98.73	99.36	99.36
Sensitivity	98	97	99	98	98	98	97
Specificity	100	100	99	100	100	100	100
PPV	100	100	99	100	100	100	100
NPV	97	96	99	97	97	97	96
F1-Score	0.99	0.99	0.99	0.99	0.99	0.99	0.99
AUC	0.998	0.997	0.998	0.992	-	1	0.986
Computational Time in sec.	0.071	0.0937	0.172	0.130	0.643	0.652	1.096



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 12. ROC Curve analysis with Corresponding AUC Values

RF and Soft Voting attained the highest accuracy, reaching 99.36%, with respective AUC scores of 0.998 and 1. However, RF requires significantly less computational power than Soft Voting, making it the most efficient among all ensemble methods tested in this research. We evaluated the inclusion of additional feature selection method RFE on the ensemble techniques and found that did not lead to significant improvements in accuracy or model performance as shown in the Table IV.

To validate the robustness of the proposed approach, a comparative performance analysis with several state-of-the-art methods is presented in Table V. Our ensemble classifier outperformed the competing methods, achieving an accuracy of 99.36%.

TABLE IV. COMPARATIVE ANALYSIS FOR ADDITIONAL RFE FEATURE SELECTION METHOD

ML Model	Accuracy with RFE	Accuracy with GA
RF	97.47	99.36
ETC	97.47	98.09
XGBoost	98.33	99.36
AdaBoost	97.47	98.73
Hard Voting	98.33	98.73
Soft voting	98.33	99.36
Stacking	98.23	99.36

TABLE V. COMPARATIVE ANALYSIS WITH STATE OF THE ART RESEARCH

Ref	Ensemble Methods	Preprocessing Technique	Dataset	Accuracy in %
[63]	Ensemble voting Ensemble Averaging	Data splitting	UCI Cleveland	Voting: 96.10 Averaging: 96.46
[24]	Ensemble Voting	Chi- square FS	UCI Cleveland	Voting: 92.11
[64]	Bagging, Boosting, Voting, Stacking,	Standardization, Random oversampling, Isolation Forest	UCI Cleveland	XGBoost, Voting, Stacking: 98.73
[16]	Weighted aging classifier ensemble	Randomization	UCI Cleveland	Weighted aging classifier ensemble 93
[20]	Soft voting	Grid search	UCI Cleveland	93.44
[22]	Stacking Ensemble	Chi- square FS	UCI Cleveland	90.8
[25]	XGBoost	Min_max scaling, FS	UCI Cleveland	94.1
Proposed Method	Bagging, Boosting, Voting, Stacking,	Standardization, Random oversampling, Isolation Forest, Data Augmentation, Feature optimization using GA	UCI Cleveland	RF, XGBoost, Soft voting, Stacking: 99.36

In this study, we evaluated the performance of several ensemble classifiers on the Cleveland heart disease dataset, demonstrating significant improvements in predictive accuracy, sensitivity, specificity, and AUC through the use of data augmentation and genetic algorithm-based feature optimization. The models achieved impressive performance metrics, showcasing their potential for improving diagnostic accuracy in medical settings.

To further assess the practical applicability of these methods, we conducted an analysis of their computational efficiency. The results showed that while models like Random Forest (RF) and XGBoost achieved high accuracy, they also required substantial processing time. Stacking, in particular, demonstrated higher computational demands compared to simpler methods like Voting, which offers a more balanced trade-off between accuracy and computational efficiency. The processing times were measured and are detailed in Table 3, which provides a clear comparison of the resource requirements for each ensemble method. These findings are particularly important for real-time clinical applications, where rapid and efficient diagnosis is essential.

V. CONCLUSION

The proposed methodology demonstrates promising performance in heart disease prediction, achieving an accuracy of 99.36% and an AUC of 1.0 under controlled conditions. However, these results should be interpreted with caution, as such perfect scores are rarely observed in real-world datasets characterized by noise and variability. Further external validation on independent datasets is essential to assess the model's generalizability and robustness. Additionally, while the model shows improved efficiency, the use of data augmentation and genetic algorithms introduces significant computational complexity, which may limit its applicability in resource-constrained environments. A balanced approach that considers the trade-offs between model complexity, computational efficiency, and predictive performance will be necessary for real-world clinical implementation. Future work will focus on validating the model across diverse datasets and optimizing it for more efficient, practical use in clinical settings.

REFERENCES

[1] K. Drożdż *et al.*, "Risk factors for cardiovascular disease in patients with metabolic-associated fatty liver disease: a machine learning approach," *Cardiovasc. Diabetol.*, vol. 21, no. 1, p. 240, Nov. 2022, doi: 10.1186/s12933-022-01672-9.

[2] M. A. Naser, A. A. Majeed, M. Alsabah, T. R. Al-Shaikhli, and K. M. Kaky, "A Review of Machine Learning's Role in Cardiovascular Disease Prediction: Recent Advances and Future Challenges," *Algorithms*, vol. 17, no. 2, p. 78, Feb. 2024, doi: 10.3390/a17020078.

[3] R. Rabiei, "Prediction of Breast Cancer using Machine Learning Approaches," *J. Biomed. Phys. Eng.*, vol. 12, no. 3, Jul. 2022, doi: 10.31661/jbpe.v0i0.2109-1403.

[4] A. Muniasamy, S. Tabassam, M. A. Hussain, H. Sultana, V. Muniasamy, and R. Bhatnagar, "Deep Learning for Predictive Analytics in Healthcare," in *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019)*, pp. 32-42, 2020.

[5] Y. Kazemi and S. A. Mirroshandel, "A novel method for predicting kidney stone type using ensemble learning," *Artif. Intell. Med.*, vol. 84, pp. 117–126, Jan. 2018, doi: 10.1016/j.artmed.2017.12.001.

[6] A. Rajdhan, A. Agarwal, M. Sai, D. Ravi, and P. Ghuli, "Heart Disease Prediction using Machine Learning," *Int. J. Eng. Res.*, vol. 9, no. 4, May 2020, doi: 10.17577/IJERTV9IS040614.

[7] M. S. Amin, Y. K. Chiam, and K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telemat. Inform.*, vol. 36, pp. 82–93, Mar. 2019, doi: 10.1016/j.tele.2018.11.007.

[8] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 117–122, 2018, doi: 10.1109/IIPHDW.2018.8388338.

[9] D. Yewale and S. P. Vijayaragavan, "Data-Driven Insights: A Genetic Algorithm Feature Optimization Approach to Heart Disease Prediction," in 2024 *International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, pp. 1–6, 2024, doi: 10.1109/IITCEE59897.2024.10467809.

[10] D. Yewale, S. P. Vijayaragavan, and M. Munot, "An optimized XGBoost based classification model for effective analysis of heart disease prediction," *1st international conference on computational applied sciences & it's applications,* p. 020019, 2023, doi: 10.1063/5.0148268.

[11] R. Biswas *et al.*, "A Robust Deep Learning based Prediction System of Heart Disease using a Combination of Five Datasets," in *2021 31st International Conference on Computer Theory and Applications (ICCTA),* pp. 223–228, 2021, doi: 10.1109/ICCTA54562.2021.9916601.

[12] P. Mahajan, S. Uddin, F. Hajati, and M. A. Moni, "Ensemble Learning for Disease Prediction: A Review," *Healthcare*, vol. 11, no. 12, p. 1808, Jun. 2023, doi: 10.3390/healthcare11121808.

[13] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," *IEEE Access,* vol. 7, pp. 81542–81554, 2019, doi: 10.1109/ACCESS.2019.2923707.

[14] A. Alqahtani, S. Alsubai, M. Sha, L. Vilcekova, and T. Javed, "Cardiovascular Disease Detection using Ensemble Learning," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Aug. 2022, doi: 10.1155/2022/5267498.

[15] V. Shorewala, “Early detection of coronary heart disease using ensemble techniques,” *Inform. Med. Unlocked*, vol. 26, p. 100655, 2021, doi: 10.1016/j.imu.2021.100655.

[16] I. D. Mienye, Y. Sun, and Z. Wang, “An improved ensemble learning approach for the prediction of heart disease risk,” *Inform. Med. Unlocked*, vol. 20, p. 100402, 2020, doi: 10.1016/j.imu.2020.100402.

[17] I. D. Mienye and N. Jere, “Optimized Ensemble Learning Approach with Explainable AI for Improved Heart Disease Prediction,” *Information*, vol. 15, no. 7, p. 394, Jul. 2024, doi: 10.3390/info15070394.

[18] S. Diwan, G. S. Thakur, S. K. Sahu, M. Sahu, and N. K. Swamy, “Predicting Heart Diseases through Feature Selection and Ensemble Classifiers,” *J. Phys. Conf. Ser.*, vol. 2273, no. 1, p. 012027, May 2022, doi: 10.1088/1742-6596/2273/1/012027.

[19] G. A. Alshehri and H. M. Alharbi, “Prediction of Heart Disease using an Ensemble Learning Approach,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 8, 2023, doi: 10.14569/IJACSA.2023.01408118.

[20] N. Chandrasekhar and S. Peddakrishna, “Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization,” *Processes*, vol. 11, no. 4, p. 1210, Apr. 2023, doi: 10.3390/pr11041210.

[21] D. Asif, M. Bibi, M. S. Arif, and A. Mukheimer, “Enhancing Heart Disease Prediction through Ensemble Learning Techniques with Hyperparameter Optimization,” *Algorithms*, vol. 16, no. 6, p. 308, Jun. 2023, doi: 10.3390/a16060308.

[22] R. R. Sarra, I. I. Gorial, R. R. Manea, A. E. Korial, M. Mohammed, and Y. Ahmed, “Enhanced Stacked Ensemble-Based Heart Disease Prediction with Chi-Square Feature Selection Method,” *Journal of Robotics and Control*, vol. 5, no. 6, 2024, doi: 10.18196/jrc.v5i6.23191.

[23] Y. Shaikh, V. K. Parvati, and S. R. Biradar, “Heart Disease Prediction using Ensemble Learning,” in *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)*, pp. 1–5, 2023, doi: 10.1109/ICAISC58445.2023.10200375.

[24] A. E. Korial, I. I. Gorial, and A. J. Humaidi, “An Improved Ensemble-Based Cardiovascular Disease Detection System with Chi-Square Feature Selection,” *Computers*, vol. 13, no. 6, p. 126, May 2024, doi: 10.3390/computers13060126.

[25] M. R. Islam, Md. Durul Hoda, Md. A. Rashid, S. Alam Suha, and M. T. Islam Miya, “Data-Driven Heart Disease Prediction by Ensemble Feature Selection and Machine Learning Techniques,” in *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pp. 575–580, 2022, doi: 10.1109/ICCIT57492.2022.10054998.

[26] Z. Noroozi, A. Orooji, and L. Erfannia, “Analyzing the impact of feature selection methods on machine learning algorithms for heart disease prediction,” *Sci. Rep.*, vol. 13, no. 1, p. 22588, Dec. 2023, doi: 10.1038/s41598-023-49962-w.

[27] K. Dissanayake and M. G. Md Johar, “Comparative Study on Heart Disease Prediction Using Feature Selection Techniques on Classification Algorithms,” *Appl. Comput. Intell. Soft Comput.*, vol. 2021, pp. 1–17, Nov. 2021, doi: 10.1155/2021/5581806.

[28] G.-I. Kim, H. Yoo, H.-J. Cho, and K. Chung, “Defect Detection Model Using Time Series Data Augmentation and Transformation,” *Comput. Mater. Contin.*, vol. 78, no. 2, pp. 1713–1730, 2024, doi: 10.32604/cmc.2023.046324.

[29] R. Gayathri, *et al.*, “Enhancing heart disease prediction with reinforcement learning and data augmentation,” *Syst. Soft Comput.*, vol. 6, p. 200129, Dec. 2024, doi: 10.1016/j.sasc.2024.200129.

[30] S. A. Ali *et al.*, “An Optimally Configured and Improved Deep Belief Network (OCI-DBN) Approach for Heart Disease Prediction Based on Ruzzo–Tompa and Stacked Genetic Algorithm,” *IEEE Access*, vol. 8, pp. 65947–65958, 2020, doi: 10.1109/ACCESS.2020.2985646.

[31] A. A. Nancy, D. Ravindran, P. M. D. Raj Vincent, K. Srinivasan, and D. Gutierrez Reina, “IoT-Cloud-Based Smart Healthcare Monitoring System for Heart Disease Prediction via Deep Learning,” *Electronics*, vol. 11, no. 15, p. 2292, Jul. 2022, doi: 10.3390/electronics11152292.

[32] H. S. Obaid, S. A. Dheyab, and S. S. Sabry, “The Impact of Data Pre-Processing Techniques and Dimensionality Reduction on the Accuracy of Machine Learning,” in *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, pp. 279–283, 2019, doi: 10.1109/IEMECONX.2019.8877011.

[33] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, “Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance,” *Technologies*, vol. 9, no. 3, p. 52, Jul. 2021, doi: 10.3390/technologies9030052.

[34] H.-J. Park, Y.-S. Koo, H.-Y. Yang, Y.-S. Han, and C.-S. Nam, “Study on Data Preprocessing for Machine Learning Based on Semiconductor Manufacturing Processes,” *Sensors*, vol. 24, no. 17, p. 5461, Aug. 2024, doi: 10.3390/s24175461.

[35] C. Yang, E. A. Fridgeirsson, J. A. Kors, J. M. Reps, and P. R. Rijnbeek, “Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data,” *J. Big Data*, vol. 11, no. 1, p. 7, Jan. 2024, doi: 10.1186/s40537-023-00857-7.

[36] H. Xu, G. Pang, Y. Wang, and Y. Wang, “Deep Isolation Forest for Anomaly Detection,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12591–12604, Dec. 2023, doi: 10.1109/TKDE.2023.3270293.

[37] J. Chen, J. Zhang, R. Qian, J. Yuan, and Y. Ren, “An Anomaly Detection Method for Wireless Sensor Networks Based on the Improved Isolation Forest,” *Appl. Sci.*, vol. 13, no. 2, p. 702, Jan. 2023, doi: 10.3390/app13020702.

[38] V. Yepmo, G. Smits, M.-J. Lesot, and O. Pivert, “Leveraging an Isolation Forest to Anomaly Detection and Data Clustering,” *Data Knowl. Eng.*, vol. 151, p. 102302, May 2024, doi: 10.1016/j.datak.2024.102302.

[39] W. Chua *et al.*, “Web Traffic Anomaly Detection Using Isolation Forest,” *Informatics*, vol. 11, no. 4, p. 83, Nov. 2024, doi: 10.3390/informatics11040083.

[40] C. Khosla and B. S. Saini, “Enhancing Performance of Deep Learning Models with different Data Augmentation Techniques: A Survey,” in *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 79–85, 2020, doi: 10.1109/ICIEM48762.2020.9160048.

[41] K. Maharana, S. Mondal, and B. Nemade, “A review: Data pre-processing and data augmentation techniques,” *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[42] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval, “Data Augmentation techniques in time series domain: a survey and taxonomy,” *Neural Comput. Appl.*, vol. 35, no. 14, pp. 10123–10145, May 2023, doi: 10.1007/s00521-023-08459-3.

[43] R. Kundu and S. Chattopadhyay, “Deep features selection through genetic algorithm for cervical pre-cancerous cell classification,” *Multimed. Tools Appl.*, vol. 82, no. 9, pp. 13431–13452, Apr. 2023, doi: 10.1007/s11042-022-13736-9.

[44] M. G. Altarabichi, S. Nowaczyk, S. Pashami, and P. S. Mashhadi, “Fast Genetic Algorithm for feature selection — A qualitative approximation approach,” *Expert Syst. Appl.*, vol. 211, p. 118528, Jan. 2023, doi: 10.1016/j.eswa.2022.118528.

[45] E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, “Enhanced Feature Selection Using Genetic Algorithm for Machine-Learning-Based Phishing URL Detection,” *Appl. Sci.*, vol. 14, no. 14, p. 6081, Jul. 2024, doi: 10.3390/app14146081.

[46] J. Chung and J. Teo, “Single classifier vs. ensemble machine learning approaches for mental health prediction,” *Brain Inform.*, vol. 10, no. 1, p. 1, Dec. 2023, doi: 10.1186/s40708-022-00180-6.

[47] S. M. Ganie and M. B. Malik, “An ensemble Machine Learning approach for predicting Type-II diabetes mellitus based on lifestyle indicators,” *Healthc. Anal.*, vol. 2, p. 100092, Nov. 2022, doi: 10.1016/j.health.2022.100092.

[48] M. Kayacı Çodur, “Ensemble Machine Learning Approaches for Prediction of Türkiye’s Energy Demand,” *Energies*, vol. 17, no. 1, p. 74, Dec. 2023, doi: 10.3390/en17010074.

[49] L. Vergni and F. Todisco, “A Random Forest Machine Learning Approach for the Identification and Quantification of Erosive Events,” *Water*, vol. 15, no. 12, p. 2225, Jun. 2023, doi: 10.3390/w15122225.

[50] W. Feng, J. Gou, Z. Fan, and X. Chen, “An ensemble machine learning approach for classification tasks using feature generation,” *Connect. Sci.*, vol. 35, no. 1, p. 2231168, Dec. 2023, doi: 10.1080/09540091.2023.2231168.

[51] E. K. Ampomah, Z. Qin, and G. Nyame, "Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement," *Information*, vol. 11, no. 6, p. 332, Jun. 2020, doi: 10.3390/info11060332.

[52] Y. O. Daddala and K. Shaik, "Cardiovascular Disease Prediction: Employing Extra Tree Classifier-Based Feature Selection and Optimized RNN with Artificial Bee Colony," *Rev. Intell. Artif.*, vol. 38, no. 2, pp. 643–653, Apr. 2024, doi: 10.18280/ria.380228.

[53] J. Montomoli *et al.*, "Machine learning using the extreme gradient boosting (XGBoost) algorithm predicts 5-day delta of SOFA score at ICU admission in COVID-19 patients," *J. Intensive Med.*, vol. 1, no. 2, pp. 110–116, Oct. 2021, doi: 10.1016/j.jointm.2021.09.002.

[54] Y. Zhang *et al.*, "A multiclass extreme gradient boosting model for evaluation of transcriptomic biomarkers in Alzheimer's disease prediction," *Neurosci. Lett.*, vol. 821, p. 137609, Jan. 2024, doi: 10.1016/j.neulet.2023.137609.

[55] S. Doki, S. Devella, S. Tallam, S. S. Reddy Gangannagari, P. Sampathkrishna Reddy, and G. P. Reddy, "Heart Disease Prediction Using XGBoost," in *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, pp. 1317–1320, 2022, doi: 10.1109/ICICICT54557.2022.9917678.

[56] E. Sevinç, "An empowered AdaBoost algorithm implementation: A COVID-19 dataset study," *Comput. Ind. Eng.*, vol. 165, p. 107912, Mar. 2022, doi: 10.1016/j.cie.2021.107912.

[57] Y. Ding, H. Zhu, R. Chen, and R. Li, "An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification," *Appl. Sci.*, vol. 12, no. 12, p. 5872, Jun. 2022, doi: 10.3390/app12125872.

[58] S. Jindal, M. Sachdeva, and A. K. Kushwaha, "Performance evaluation of machine learning based voting classifier system for human activity recognition," *Kuwait J. Sci.*, Jun. 2022, doi: 10.48129/kjs.splml.19189.

[59] S. Hadhri, M. Hadiji, and W. Labidi, "A voting ensemble classifier for stress detection," *J. Inf. Telecommun.*, vol. 8, no. 3, pp. 399–416, Jul. 2024, doi: 10.1080/24751839.2024.2306786.

[60] P. W. Khan, Y. C. Byun, and O.-R. Jeong, "A stacking ensemble classifier-based machine learning model for classifying pollution sources on photovoltaic panels," *Sci. Rep.*, vol. 13, no. 1, p. 10256, Jun. 2023, doi: 10.1038/s41598-023-35476-y.

[61] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, and R. Kashef, "A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients," *Decis. Anal. J.*, vol. 7, p. 100242, Jun. 2023, doi: 10.1016/j.dajour.2023.100242.

[62] M. Ali, M. N. Haider, S. A. Lashari, W. Sharif, A. Khan, and D. A. Ramli, "Stacking Classifier with Random Forest functioning as a Meta Classifier for Diabetes Diseases Classification," *Procedia Comput. Sci.*, vol. 207, pp. 3459–3468, 2022, doi: 10.1016/j.procs.2022.09.404.

[63] B. Adhikari and S. Shakya, "Heart Disease Prediction Using Ensemble Model," in *Proceedings of Second International Conference on Sustainable Expert Systems*, vol. 351, pp. 857–868, 2022, doi: 10.1007/978-981-16-7657-4_69.

[64] D. Yewale, S. P. Vijayaragavan, and V. K. Bairagi, "An Effective Heart Disease Prediction Framework based on Ensemble Techniques in Machine Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 2, 2023, doi: 10.14569/IJACSA.2023.0140223.