Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

An Innovative Machine Learning Framework for Cardiovascular Disease Detection

Jayasudha T.[1], Uma Rani R.[2]

[1]Research Scholar, PG & Research Department of Computer Science, Sri Sarada College for Women (Autonomous), Salem, India.

[2]Principal (Retired), Sri Sarada College for Women (Autonomous), Salem, India.

E-mail: [1]jayasudha.sasmca@gmail.com, [1]jayasudhat.mca@gmail.com, [2]drrumarani66@gmail.com

Abstract

The most pivotal condition affecting human health is cardiac disease (CVD). Early detection of CVD can help prevent or mitigate its impact, potentially lowering mortality rates. Machine learning models are employed to identify CVD risk factors. To enhance CVD detection, we propose a robust framework by utilizing a variety of feature selection techniques to identify key predictive traits, using K-Fold cross-validation to prevent overfitting and model selection, and applying several novel ensemble classification methodologies. Real-time data were collected from a private hospital in Salem, and benchmark combined datasets were used for cardiovascular disease detection. A feature-type-based technique is used for handling missing values, and the Z-score technique is utilised for outlier handling. The SMOTE method is used to balance the imbalanced class. Three feature selection techniques, i.e., Pearson Correlation Coefficient, Recursive Feature Elimination, and Random Forest Feature Importance, are used to select the best attributes. Innovative ensemble classifiers like Bagging-Boosting Stacked Ensemble (BBSE), Heterogeneous Soft Voting Ensemble (HSVE), Feature-Augmented Heterogeneous Stacking (FAHS), Heterogeneous Bootstrap-Ensemble (HBE), and Heterogeneous Sequential Boosting (HSB) are created by combining multiple classifiers. The confusion matrix, accuracy, F1 score, recall, precision, and ROC were employed to measure performance. In a real-time medical dataset, the FAHS scored the highest accuracy of 92.18% without feature selection and the K-Fold CV methods. After applying the attribute selection methods and the K-fold CV approach, the FAHS model with the random forest feature importance technique scored the highest accuracy of 96.09%. In the benchmark dataset, FAHS scored the highest accuracy of 88.67% without feature selection and K-Fold CV. After applying the feature selection approaches and K-fold CV technique, the FAHS classifier with the random forest feature importance strategy scored the highest accuracy of 94.09%. Cardiovascular disease is a major global health problem, requiring correct and early detection. This study assesses different AI models, including FAHS, HSB and blended architectures, on a real- world medical dataset. The experimental output describes that the hybrid FAHS type exceeds traditional classifications, achieving 96.8%validity, 95.5% precision, 96.2% recall, and an AUC of 0.97. These findings illuminate the potential of ensemble learning frameworks to enhance predictive interpretability, accuracy, and scalability in CVD detection for practical healthcare implementation. In the real-time dataset, accuracy was improved from 92.18% to 96.09%. On the benchmark dataset, accuracy was improved from 88.67% to 94.09%. The random forest feature importance method with the FAHS combination scored the highest

DOI: https://doi.org/10.36548/jiip.2025.4.001
Received: 28.07.2025, received in revised form: 01.09.2025, accepted: 18.09.2025, published: 07.10.2025

accuracy on both datasets. The outcomes are shown individually to provide comparisons. We may conclude from the outcome analysis that our suggested models provided the highest accuracy. In the future, these models will be very beneficial in detecting CVD with high accuracy.

Keywords: Cardiovascular Disease, Feature Selection, K-Fold Cross-validation, Feature-Augmented Stacking, Sequential Boosting.

1. Introduction

Because of its high prevalence and death rates, cardiovascular disease (CVD) continues to be one of the major global health threats, severely affecting healthcare systems. The World Health Organization (WHO) estimates that CVD causes 17.9 million deaths a year, or nearly one-third of all deaths worldwide, with that number expected to reach 30 million by 2040 [1-4]. Although cardiovascular problems are most prevalent in middle-aged or older men, research indicates that they can also affect younger people, underestimating their widespread impact [5-8].

Risk factors for the detection of CVD, such as gender, age, type of pain, blood pressure, heart rate, blood sugar, exercise, diet, and family history, must be investigated quantitatively [9]. To achieve promising detection accuracies, the current final works have applied machine learning (ML) and its types, such as decision trees and support vector machines, to these challenging indicators [10–13]. However, the majority of earlier methods are frequently limited by the interpretability of features, imbalanced datasets, and inability to scale across different populations.

This study is innovative because it goes beyond traditional machine learning-based detection frameworks by utilizing ensemble learning architecture that can extract complex and non-linear relationships from clinical and lifestyle data. Aside from earlier approaches that mainly respond to cross-domain validation, real-world applicability, and handcrafted attribute extraction, this study highlights which architectures are better at generalizing to diverse populations by comparing evaluations of several AI models. The proposed framework aims to close the gap between theoretical detection models and real-world, scalable healthcare applications by combining interpretability mechanisms and placing a strong emphasis on deployment feasibility.

1.1 Cardiovascular Disease Types

Few types of CVD types:

- Coronary artery disease (CAD): Heart blood vessel blockage.
- Hypertension: Blood pressure.
- Heart failure: Weak heart can’t inject blood.
- Arrhythmias: Uncommon types of heartbeats.
- Stroke: Disturbance of brain blood function.
- Peripheral Artery Disease (PAD): Blood vessel narrowing in the extremity.
- Congenital Heart Defects: Heart defects that occur at birth.
- Rheumatic heart disease: Rheumatic fever-induced valve decline.
- Cardiomyopathy: Heart’s muscles are weak.
- Aortic dissection or aneurysm: Ruptured aorta or a weak.

- Venous Thromboembolism (VTE): Clots in the veins or lungs.
- Endocarditis: If heart inner layer affects mean heart will be infected

1.2 Research Contribution

The following are a few of the research's main contributions:

- Real-time and benchmark datasets are used for detection.
- Pre-processing data by eliminating redundant information, handling missing data and outliers, and class balancing.
- Identify the most essential attributes to detect CVD.
- Identify the best feature selection method.
- The 5-fold CV technique is used to address overfitting issues.
- Combining bagging and boosting strategies with traditional models for effectively classifying disease.
- Identify the best classification method.
- Improving old manual systems.
- Improving accuracy.
- Enhancing effectiveness and efficiency.
- The proposed framework's outcome is evaluated against the outcomes of the previous studies.

Lack of Real-Time Data and Imbalance

Traditional ML models require large, high-quality labeled datasets and often perform poorly on imbalanced clinical data, leading to the misclassification of high-risk patients.

Limited Interpretability

Many existing ML approaches act as “black boxes,” making it difficult for clinicians to understand or trust predictions.

Sensitivity to Noise and Non-Linearity

Conventional models may fail to capture complex, non-linear interactions among risk factors and are often affected by missing or noisy data, reducing their reliability in real-time clinical applications

2. Literature Survey

This work [6] developed a new diagnostic method using the Cleveland dataset from the UCI repository. Once missing values were handled, the DT, SVM, RF, and KNN were used for classification. The KNN achieved a maximum accuracy of 87%. R. Aggrawal et al. [8] described a sequential feature selection strategy for detecting mortality events in heart disease patients. Several machine-learning techniques were used, including KNN, RF, SVM, DT, and GBC. According to the outcomes, the random forest classifier scored the highest accuracy of 86.67% with the K-fold CV and sequential feature selection approach.

Spencer et al. [14] used four alternative feature selection methods to forecast heart disease: principal component analysis, ReliefF, Chi-squared testing, and symmetrical uncertainty. The highest accuracy of 85.0% was observed when the Chi-squared variable selection was integrated with the Bayes net classifier and 10-fold CV technique. Takci [15] employed four optimal variable selection techniques and twelve classification techniques from various categories to determine cardiac attacks. Without variable selection, the highest accuracy value was 82.59%, and with variable selection, it rose to 84.81% using naive Bayes and linear SVM.

The Framingham Heart Disease and CVD datasets were used for HD analysis [17]. Important traits were chosen using the ANOVA-F test. When using all features, the Perceptron model had the highest accuracy on the CVD dataset (0.73), and the Linear SVC model had the highest accuracy on the Framingham dataset (0.66). After feature selection was applied, the accuracy increased to 0.74 for the CVD dataset using SVC and to 0.71 for the Framingham dataset using Perceptron. Akua Sekyiwaa et al. [18] used various machine learning methods to forecast cardiac disease. They applied models such as KNN, SVM, Logistic Regression, and ANN to the UCI combined heart disease dataset. They tuned the models using GridSearchCV. The KNN provided the highest accuracy of 87% among these.

Snigdha Datta [19] used a logistic regression model to predict cardiac disease. The UCI Heart Disease data was used in the study to make predictions. The model was validated using 10-fold cross-validation to ensure consistent and trustworthy outcomes. To find significant features, the author additionally employed tests such as the Wald and Likelihood-Ratio. The accuracy of the logistic regression technique was 81%. Using the 303 records in the Cleveland dataset, Abdar [20] discovered an enhanced decision tree approach that may be used to derive rules for prognosticating heart problems, CAD, and CVD. They proposed a C5.0 method with an accuracy of roughly 85.33%. The most recognized factors for forecasting heart disease is a mixture of characteristics, including trestbps, restecg, thalach, slope, oldpeak, and cp. This model could be utilised to determine the variables affecting heart patients.

2.1 Research Gap

The following significant research gaps have been found in the field of CVD detection based on the examined literature:

- The research gap in cardiovascular disease detection is that the availability of real-time datasets is limited, and the accuracy needs to be enhanced.
- Algorithm selection also plays a vital role in cardiovascular illness detection.
- The following advanced ensemble methods are still not well studied in CVD detection:
    1. Feature-augmented stacking with various models to improve feature learning.
    2. Sequential boosting with various models to improve prediction refinement.
    3. Bootstrapped ensembles using mixed models for improved accuracy and stability.
    4. Combining bagging and boosting techniques to take advantage of both variance and bias reduction strengths.

This study fills these gaps by proposing a unique paradigm that aims to improve CVD detection accuracy and reliability.

3. Proposed Workflow

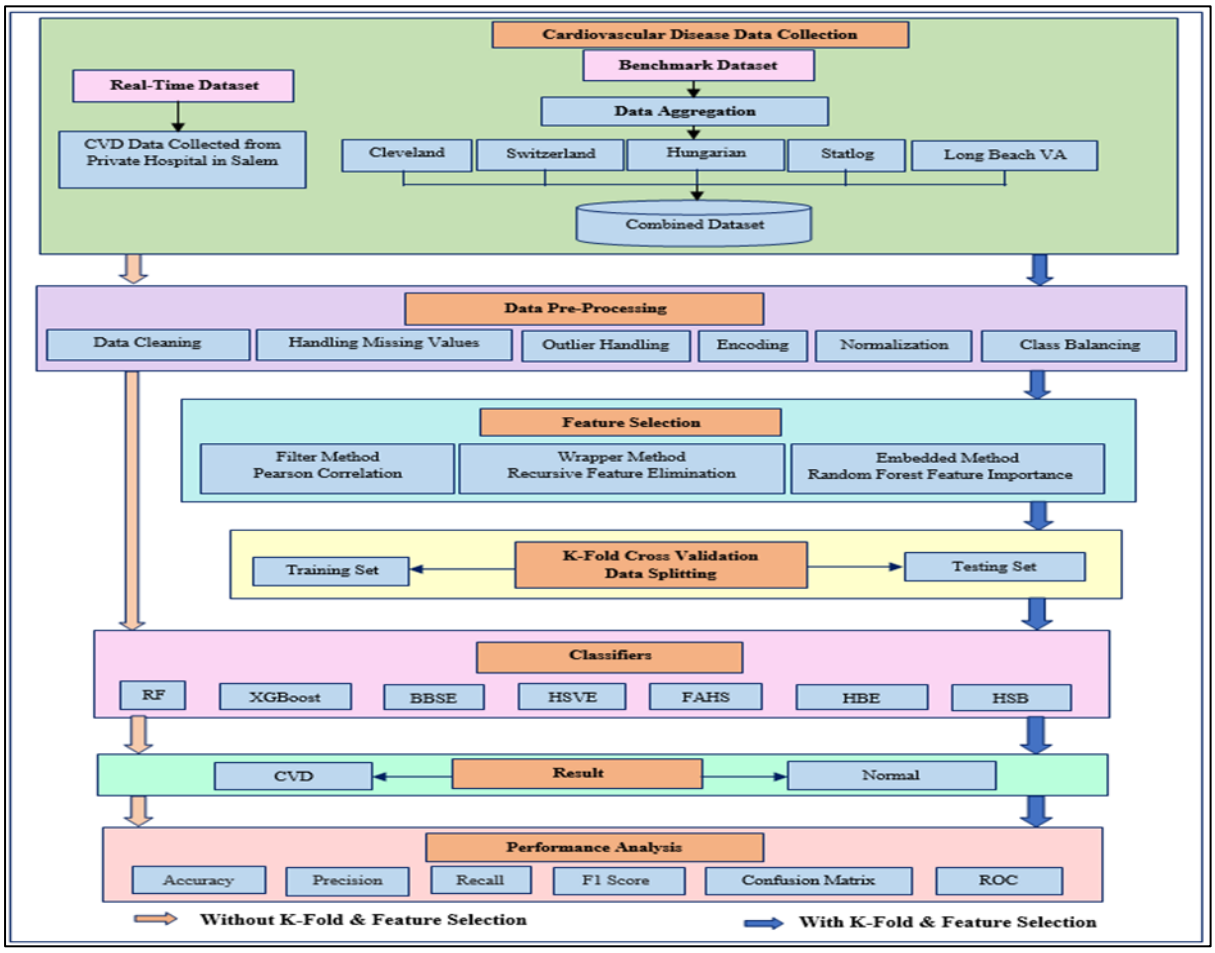

Figure 1. Novel CVD Detection Framework

The method of differentiating the work by integrating feature selection with a different ensemble learning framework is mainly distinguished for cardiovascular disease (CVD), increasing both predictive accuracy and clinical interpretability. Learning is easier when relying on a single crispification of generic ML models. This model describes different types of beginners with a meta-learner to uncover the complex nonlinear relationships through the clinical factors. The content utilized comprises real-time tolerant records, including lifestyle differences, medical variables, and demographics, to facilitate robust evaluation across diverse cardiac profiles and demonstrate superior performance compared to existing methods.

Figure 1 depicts the flow of CVD detection strategies. The first step is collecting cardiovascular disease data. After collecting the data, the next stage is preprocessing, which includes data cleaning, imputation, outlier handling, encoding, normalization, and class balancing. The best features are then chosen using feature selection techniques. Later, RF, Boost, BBSE, HSVE, FAHS, HBE, and HSB are applied for CVD detection with the K-fold CV technique. After using these algorithms and strategies, we compare the outcomes and present our conclusions. This framework initially performs detection without variable selection

and K-Fold CV, and then it performs detection with variable selection and K-Fold CV. This research explores the impact of the attribute selection strategy, K-Fold CV approach, and ensemble classification methods on improving cardiovascular disease detection.

3.1 CVD Dataset

The first step in this investigation is collecting both real-time data and open-source UCI cardiovascular disease data. For the real-time data, patient details were collected from a private hospital in Salem. For the benchmark dataset, combinations of five different datasets are used for cardiovascular detection.

3.2 Data Preprocessing

It is impossible to directly develop machine learning models from real-world data, which typically contains noisy information and missing values and may be in an unfavorable format. Data preprocessing improves a machine learning technique's accuracy and efficiency. It is necessary to clean and prepare the data for the model.

3.2.1 Data Cleaning

The main motive is to upgrade the standard and dependability of the data, ensuring that it is precise, consistent, comprehensive, and appropriate for modeling, or decision making. In this research, data cleaning was done to find and fix mistakes, inaccuracies, invalid entries, inconsistencies, and unnecessary portions of the data. As a result, the data quality is increased, enhancing its utility. In this study, data cleaning was done to find and fix mistakes, inconsistencies, inaccuracies, duplicate entries, and unnecessary portions of the data. As a result, data quality is enhanced, increasing its utility.

3.2.2 Handling Missing Values

```
Input: A Dataset with missing values.
Algorithm:
 For each feature i in the Dataset:
   If i has missing values:
      If i is binary:
         Compute the mode of i.
         Replace all missing values in i with the mode.
      Else if i is numerical:
         Compute the median of i, excluding missing values.
         Replace all missing values in i with the median.
      Else if i is categorical:
         Replace all missing values in i with "Others".
Return a cleaned dataset.
End Algorithm
Output: Dataset with no missing values.
```

Figure 2. Imputation Pseudocode

Dealing with missing values involves handling incomplete or missing data points in a dataset. Various factors, including human mistakes, system faults, and problems with data

collection, can cause missing values. Many machine learning algorithms cannot directly handle missing data; hence, it is crucial to manage them correctly. This study employs a feature-type-based methodology to handle missing values (Fig. 2).

The most prevalent value fills in the missing binary values. Data consistency is maintained by substituting the median for numerical missing values and "Others" for categorical ones. These imputations guarantee appropriate management while preserving the integrity of the dataset.

3.2.3 Handling Outliers

Outliers are unusual data points that significantly depart from the norm. They can be much higher or lower than most other values and often stand out as anomalies or extreme values. Handling outliers is crucial for improving model accuracy, data quality, and the validity of insights derived from the data. To preserve data integrity, numerical outliers are detected using the Z-score approach and replaced with the median of non-outliers. The z-score is a statistical metric for detecting outliers within a dataset by quantifying how far away a data instance is from the mean.

3.2.4 Encoding

Encoding, in data preprocessing, refers to transforming categorical data into a number-based format that models can use efficiently. This conversion is essential because many machine learning models need numerical data for prediction. In this study, target encoding is used to encode the categorical data. Target encoding with cross-validation is a leakage-free technique for encoding categorical variables that substitutes the target variable's mean, which is calculated solely from the training fold and not the row itself for each category.

3.2.5 Data Normalization

In data analytics, data normalisation is a pre-processing method that scales and transforms data into a consistent range or distribution. It ensures that no feature overshadows any other, regardless of scale or importance. In this work, normalisation is done via min-max scaling. The min-max scaling feature scaling approach converts the features to a specific range, usually 0 to 1.

3.2.6 Class Balancing

Class balancing is resolving imbalanced datasets in which the number of samples in one or more classes is substantially less than in the other class (es). It aims to provide a roughly equal distribution of samples across all classes in the dataset, which improves machine learning models' performance by minimising bias towards the majority class. In this research, the Synthetic Minority Oversampling Technique (SMOTE) is applied for class balancing.

SMOTE is a data augmentation method that creates synthetic samples for the underrepresented class to balance unbalanced datasets. It generates new data points by interpolating current minority class samples and their closest neighbors.

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Where:

- The Euclidean distance between the two points is denoted by d (Eq. 1).
- n is the number of dimensions.
- The coordinates of the two points in each n dimension are denoted by x_i and y_i.

A lower Euclidean distance indicates that the two points are more comparable since they are closer. A smaller Euclidean distance between neighbors produces synthetic samples that are more accurate because they are closer to the original data point.

3.3 Feature Selection

An attribute is one that either makes the problem harder to solve or makes it easier. Variable selection is the process of choosing essential traits for the model. By using the most important data, feature selection can limit the predictors for the model and help prevent high variance problems. The filter-based method determines the score derived from statistical measures and their dependence on the class label for each characteristic [14, 15, and 21]. The wrapper approaches utilize a particular learning algorithm to identify the characteristics best suited for a given dataset. Embedded feature selection picks key characteristics during model training. The benefits of filter and wrapper approaches are combined in the embedded technique. This research employs filter, wrapper, and embedded predictor selection approaches to find the most crucial characteristics.

3.3.1 Pearson Correlation Coefficient Method

The Pearson correlation coefficient method is a filter-based variable selection technique. The relationship between the continuous attributes and the class feature is discovered using the pearson correlation approach. Its value falls between -1 and 1, where a total negative linear correlation is denoted by -1, no correlation is denoted by 0, and a total positive correlation is denoted by +1.

Steps of the Pearson Correlation Coefficient Method

Step 1: Calculation of Variable Means: The Pearson correlation method's initial action is to discover the mean of each variable (X, Y) separately (Equation 2)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \, , \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{2}$$

Step 2: Computation of the Covariance Component: For the numerator, the sum of the products of the deviations of each sample from their respective means was computed (Equation 3).

$$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{3}$$

Step 3: Standard Deviation Calculation for Variables: The denominator was then determined by taking the square root of the total of the squared deviations for each variable from its mean (Equation 4).

$$\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})} \sqrt[2]{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

Step 4: Computation of Pearson's Correlation Value: The Pearson correlation coefficient, calculated by dividing the numerator by the denominator and yielding a number between -1

and 1 (Equation 5), shows the strength and direction of the linear relationship between X and Y.

$$r = \frac{Numerator}{Denominator} \tag{5}$$

Here, r: Pearson correlation coefficient. x_i: Value of the i^{th} data instance of feature x. y_i: Value of the i^{th} data instance of the target feature y. $\bar{x}$: Mean of the values in x. $\bar{y}$: mean of the values in y. n: Number of data instances.

3.3.2 Recursive Feature Elimination (RFE)

The recursive feature elimination is a wrapper-based attribute selection method. Recursive feature elimination involves iteratively fitting a model and eliminating the least important attributes according to the model's coefficients or variable relevance values. The RFE can lessen overfitting and enhance the model's generalisability by removing superfluous or unnecessary features.

Steps of RFE

Step 1: Train SVM Model: Using every feature in the CVD dataset, a linear Support Vector Machine was trained. The decision-making function is (Equation 6):

$$f(x) = w^T x + b \tag{6}$$

Where:

- w = feature coefficients (weights)
- x = input features
- b = bias

Step 2: Calculate Feature Importance: The absolute value of each feature's coefficient indicates how important it is (Equation 7):

$$Importance_i = |w_i| \tag{7}$$

Where w_i is the coefficient of the i^{th} feature.

Step 3: Rank Features: Every feature was rated according to its determined importance scores.

Step 4: Remove Least Important Feature: The feature that received the lowest significance score was removed.

Step 5: Retrain and Repeat: Steps 1 through 4 were repeatedly performed until the desired number of characteristics remained.

3.3.3 Random Forest Feature Importance (RFFI)

Random Forest Feature Importance is an embedded attribute selection technique. To assess a feature's importance, each tree in the random forest can use its potential to increase the purity of its leaves.

Steps of RFFI

Here is a detailed explanation of the Gini-based feature importance computation that the random forest truly employs.

Step 1: Impurity Calculation for Parent Node: Before the split, the parent node's Gini impurity was calculated (Equation 8).

The ΔGini indicates how much the split reduced the impurity

$$Gini\ (\text{Parent}) = 1 - \sum_{i=1}^{c} P_i^2 \quad (8)$$

Where:

- The number of classes is C.
- Pi is the likelihood that an observation belongs to class I.

Step 2: Calculating Subnode Impurities After Split: The Gini Impurity for the left and right immediate subnodes was then computed after the data were divided using a feature (Equation 9).

$$Gini\ (\text{Node}) = 1 - \sum_{i=1}^{c} P_i^2 \quad (9)$$

Step 3: Calculating Weighted Gini Impurities: Using weights determined by the number of samples in each child node, the weighted sum of the Gini impurity of the immediate subnodes was calculated (Equation 10).

$$Weighted\ Gini = \frac{N_1}{N}\ Gini\ (Node\ 1) + \frac{N_2}{N}\ Gini\ (Node\ 2) \quad (10)$$

N1: The count of samples in Node 1 after the split.

N2: The count of samples in Node 2 after the split.

N: The overall count of samples in the parent node before the split (N=N1+N2).

Step 4: Calculation of Impurity Reduction: Lastly, the reduction in impurity, ΔGini, was obtained by subtracting the weighted total of the Gini impurity of the direct subnodes from the Gini impurity of the parent node (Equation 11).

$$\Delta\text{Gini} = \text{Gini}_{\text{ before split}} - \text{Gini}_{\text{ after split}} \quad (11)$$

Step 5: Feature Importance Aggregation:

$$\text{Feature Importance} = \sum(\Delta\text{Gini for each node where features are used}) \quad (12)$$

Add all the feature's Gini Impurity (ΔGini) reductions across all tree splits to determine its importance (Equation 12).

3.4 K-fold Cross-validation Technique

Cross-validation is a statistical technique utilized to evaluate a predictive model's performance and estimate its test error to improve generalization. The concept of cross-

validation is to divide the sample into several sets. There are numerous varieties of cross-validation. This research uses a K-fold cross-validation approach to validate the results. The K-fold cross-validation method divides the dataset into K folds. Every iteration uses K-1 folds for training, while the remaining fold is used for validation. By iterating this procedure K times, each data point is guaranteed to appear exactly once in a validation set. In this study, 5-fold cross-validation is used to reduce overfitting.

Five-fold Cross-validation Steps

Step 1: The training dataset is divided into five equal subsets (folds): f1, f2, f3, f4, and f5.

Step 2: For i=1 to 5

- The i-th fold is chosen as the validation set.
- The training set consists of the remaining 5−1 folds.

Step 3: Use the training set to train the ML model.

Step 4: Assess the model's accuracy using the validation set.

Step 5: For all five folds, repeat steps 3 and 4.

Step 6: Calculate the final accuracy by taking the average of the five iterations' accuracies.

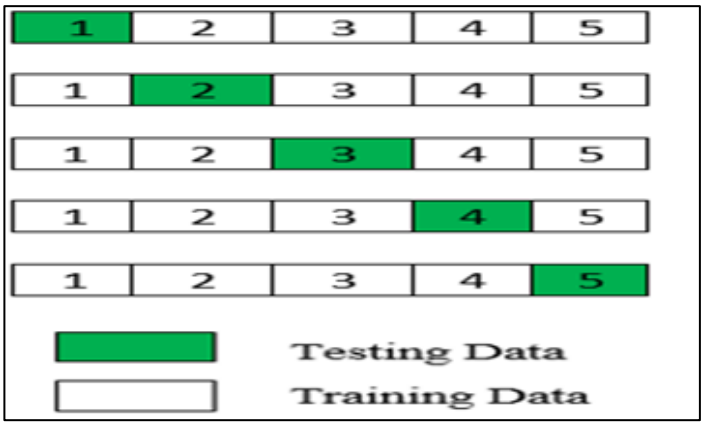

Figure 3. Five-fold Cross-Validation

As shown in Figure 3, the dataset is separated into five folds, each approximately equal in size. The cross-validation will employ one-fold for testing and four for training.

3.5 Classification Algorithms

3.5.1 Bagging-Boosting Stacked Ensemble (BBSE)

A Bagging-Boosting Stacked Ensemble is a sophisticated ensemble machine-learning method that improves model performance by fusing the ideas of boosting and bagging. This model uses the bagging technique (Extra Trees) to reduce variance and the boosting technique (LightGBM) to reduce bias. Through the integration of bagging, boosting, K-fold CV and a feature-augmented stacking strategy, this model presents a unique approach that sets it apart from earlier methods for CVD detection. The BBSE pseudocode is represented in Figure 4.

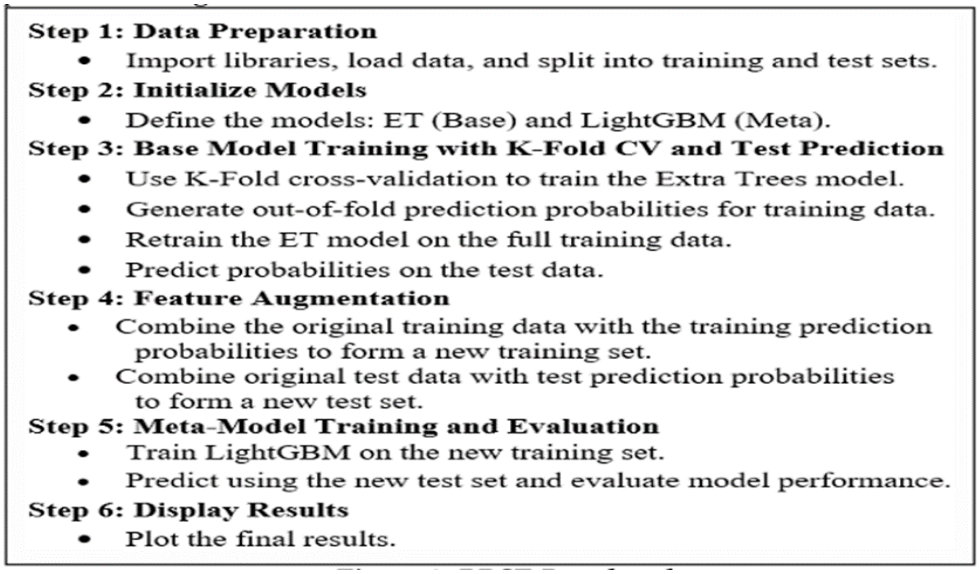

Figure 4. BBSE Pseudocode

Steps of BBSE

The dataset was initially loaded and afterward divided into training and test sets. To produce out-of-fold prediction probabilities for the training data, the Extra Trees model is trained using 5-fold cross-validation. It is then retrained on the entire training set to produce prediction probabilities for the test data. LightGBM is trained by integrating the base training features with the ET training probabilities, often known as meta-features. After that, LightGBM uses the base test features and ET test probabilities to produce end detection. The efficiency of the model is then assessed by visualising the outcomes. Here, ET serves as the underlying model, and LightGBM serves as the meta-model.

3.5.2 Heterogeneous Soft Voting Ensemble (HSVE)

The soft voting ensemble averages the estimated probability of multiple models instead of choosing the majority vote. This approach allows each model to contribute based on its degree of confidence, improving the accuracy and smoothness of detection. By combining heterogeneous classifiers, K-fold CV, and soft voting ensemble tactics, this method offers a novel and comprehensive approach that is not often found in previous related studies. Figure 5 displays the step-by-step procedure of the heterogeneous soft voting ensemble approach, which utilizes the GNB, LR, SVM, and CatBoost models. This combination combines probabilistic-based (GNB), linear-based (LR), Margin-based (SVM), and boosting-based (CatBoost) models.

Steps of HSVE

The initial step is data preparation. Next, the models are initialised. All models are combined in a soft voting classifier to increase accuracy. To assess the performance of the soft voting classifier, 5-fold cross-validation is used on the training set. The soft voting classifier is retrained with the complete training dataset following cross-validation. The final assessment metrics are then presented based on the predictions made by this final trained model on the test set.

Step 1: Data Preparation
- Import libraries, load data, and split into training and test sets.

Step 2: Initialize Models
- Define the models: GNB, LR, SVM, and CatBoost.

Step 3: Create Soft Voting Classifier
- Combine all models into an ensemble using the soft voting method.

Step 4: Apply K-Fold CV on Training Set
- For each fold:
  - ✓ Train the soft voting classifier on the training split.
  - ✓ Validate on validation split and calculate metrics.
- Compute average metrics across folds.

Step 5: Final Model Training and Testing
- Retrain the Soft Voting model on the full training set.
- Predict on the test set and evaluate performance metrics.

Step 6: Display Results
- Plot the final results.

Figure 5. HSVE Pseudocode

3.5.3 Feature-Augmented Heterogeneous Stacking (FAHS)

Feature-augmented stacking is an ensemble learning strategy that improves on the traditional stacking method by training the meta-learner with both the original input features and the predictions from the base models. This technique improves performance over standard stacking by assisting the meta-learner in capturing more intricate patterns, minimising information loss, and correcting biases. This approach presents a fresh model that is very different from other works on heart disease detection by merging heterogeneous classifiers, K-fold CV, and feature-augmented stacking approaches. Figure 6 demonstrates the algorithmic steps of the FAHS model. Five different models: LR, Extra Trees (ET), SVM, KNN, and XGBoost were used in FAHS. This combination aggregates linear-based (LR), tree-based (ET), Margin-based (SVM), distance-based (KNN), and boosting-based (XGBoost) models.

Steps of FAHS

The process starts with setting up the dataset. Model initialisation is the next phase. To train each base model, 5-fold cross-validation was used. For the training set, out-of-fold prediction probabilities were gathered. The models were then used to estimate probabilities on the test set after being retrained on the entire training set. The original features are blended with these probabilities to produce improved datasets. To get final predictions, the meta-model XGBoost is trained using the improved training data and evaluated using the improved test data. After that, the model's performance is assessed, and the outcomes are displayed.

Step 1: Data Preparation
- Import libraries, load data, and split into training and test sets.

Step 2: Initialize Models
- Define the base models: LR, ET, SVM, and KNN.
- Define the meta-model: XGBoost.

Step 3: Base Model Training with K-Fold CV and Test Prediction
- For each model:
  - ✓ Use K-Fold CV to train the base models.
  - ✓ Get out-of-fold prediction probabilities for training data.
  - ✓ Retrain the model on full training data.
  - ✓ Predict probabilities on the test data.

Step 4: Feature Augmentation
- Combine the original training data with the training prediction probabilities to form a new training set.
- Combine original test data with test prediction probabilities to form a new test set.

Step 5: Meta-Model Training and Evaluation
- Train XGBoost on the new training set.
- Predict using the new test set and evaluate model performance.

Step 6: Display Results
- Plot the final results.

Figure 6. FAHS Pseudocode

3.5.4 Heterogeneous Bootstrap-Ensemble (HBE)

The goal of the HBE is to increase model accuracy by using predictions from many models, each trained on distinct bootstrapped data, through a K-Fold Cross-Validation architecture. The ensemble improves generalisation and model diversity through a soft voting ensemble.

Step 1: Data Preparation
- Import libraries, load data, and split into training and test sets.

Step 2: Initialize Models with Bagging
- Create Bagging Classifier for: QDA, SVM, KNN.
- Define Random Forest (no manual bagging).

Step 3: Create Soft Voting Ensemble
- Combine the four trained models into a soft voting classifier.

Step 4: Perform K-Fold Cross-Validation
- For each fold:
  - ✓ Train the voting ensemble on the training fold.
  - ✓ Evaluate accuracy on the validation fold.
- Compute average accuracy across all folds.

Step 5: Final Model Training and Testing
- Train the voting ensemble on the full training set.
- Predict on the test set.
- Evaluate performance

Step 6: Display Results
- Plot the final results.

Figure 7. HBE Pseudocode

This HBE boosts the model's stability and helps to reduce variance. This approach introduces an innovative model that is very different from other research on CVD detection by merging bagging-based diverse classifiers, K-fold CV, and soft voting ensemble approaches. Figure 7 illustrates the pseudocode of the heterogeneous bootstrap-ensemble model. Four different models: Quadratic Discriminant Analysis (QDA), SVM, KNN, and RF, were used in HBE. This combination incorporates probabilistic-based (QDA), Margin-based (SVM), distance-based (KNN), and tree-based (RF) models.

Steps of HBE

Data preparation is the first phase, while model definition is the second. Bagging classifiers are built for QDA, SVM, and KNN, and RF is directly incorporated. The soft voting classifier is used to merge these models. To get the average accuracy, the ensemble is tested using K-fold cross-validation. The findings are then displayed once it has been trained on the entire training set and tested on the test set.

3.5.6 Heterogeneous Sequential Boosting (HSB)

In sequential boosting, several models are trained one after the other, each one concentrating on the mistakes made by the one before it. The primary objective is to increase the model's accuracy by lowering bias and variance through iterative adjustments. By integrating the advantages of several models into a sequential boosting framework, heterogeneous sequential boosting enhances prediction performance and produces more robust learning on complicated datasets, better error correction, reduced bias, and improved generalisation. This strategy differs from traditional methods in that it combines differential classifiers, K-fold CV, and sequential boosting methodologies to create a unique and comprehensive approach. Figure 8 demonstrates the pseudocode of a heterogeneous sequential boosting model, which utilizes LR, DT, SVM and XGBoost. This combination integrates linear-based (LR), tree-based (DT), Margin-based (SVM), and boosting-based (XGBoost) models.

Step 1: Data Preparation
- Import libraries, load data, and split into training and test sets.

Step 2: K-Fold Setup and Weight Initialization
- For each training fold:
  - ✓ Let N be the number of training samples.
  - ✓ Assign equal weight w = 1/N to each instance.

Step 3: Boosting within Each Fold and Evaluation
- Initialize all models (LR, DT, SVM and XGBoost).
- For each model in the sequence (LR → DT → SVM→ XGBoost):
  - ✓ Train on the training fold using current sample weights.
  - ✓ Predict on the same training fold.
  - ✓ Update weights (↑ for misclassified, ↓ for correct classification).
  - ✓ Normalize weights and pass new weights to the next model.
- After training the full model sequence:
  - ✓ Combine predictions using a weighted ensemble.
  - ✓ Predict on the validation fold.
- Calculate the average performance across all folds.

Step 4: Final Model Training and Testing
- Retrain the full model sequence on the complete training set.
- Predict on the test set using the final ensemble.
- Evaluate performance metrics.

Step 5: Display Results
- Plot the final results.

Figure 8. HSB Pseudocode

Steps of HSB

The first step in the procedure is data preparation. Then assign equal weights to each fold of the 5-fold cross-validation and train the models in a sequential manner (LR → DT →

SVM → XGBoost). After every model, sample weights are changed to highlight instances of incorrect classification. Finally, on the test fold, the predictions from each model are combined using a weighted sum, and the model's performance is assessed using the relevant metrics. Throughout the cross-validation procedure, these actions are performed for every fold. The performance outcomes are provided once all models have been retrained on the entire training dataset and predictions have been made on the test set.

3.6 Results and Discussion

The suggested model produces a binary result at the end of the classification stage that shows if a person is expected to have cardiovascular disease or not. A prediction label of '0' indicates the absence of CVD, while a label of '1' indicates the existence or high risk of CVD. This output represents the model's judgment about the patient's cardiovascular health and is based on the patterns discovered in the training data.

3.7 Performance Analysis

Evaluation metrics assess the efficacy of a statistical or machine learning model. A model can be tested using a wide variety of evaluation indicators. In this stage, the findings of the classifiers are displayed and compared among them. The final result will be based on the classifier with the highest accuracy on the given dataset. The confusion matrix is the foundation for these metrics. It is a two-by-two matrix that compares the algorithm's predicted class values with the actual class values. Table 1 illustrates the confusion matrix.

Table 1. Confusion Matrix

Actual Class	**Predicted Class**	
	CVD = 1	**CVD = 0**
CVD = 1	TP	FN
CVD = 0	FP	TN

The confusion matrix can be used to draw the following conclusions:

- True Positive (TP) prediction
- True Negative (TN) prediction
- False Positive (FP) prediction
- False Negative (FN) prediction

Consequently, the evaluation metrics can be outlined as follows:

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / \text{The total number of predictions} \quad (13)$$

$$\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive}) \quad (14)$$

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative}) \quad (15)$$

F1 Score = 2 * (Precision * Recall) / (Precision + Recall) (16)

- The percentage of accurately categorised outcomes with correct positives and negatives is known as accuracy (Eq. 13).
- The percentage of accurately classified outcomes that are positively classified is known as precision (Eq. 14).
- The percentage of positive instances that are correctly classified is referred to as recall (Eq. 15).
- The percentage of correctly predicted positive instances for all optimistic forecasts is the F1 Score (Eq. 16).

4. Dataset Description

4.1 Real-Time Dataset Details

For this study, real-time patient data were collected from a private hospital in Salem, India, covering the period between 2019 and 2023. The dataset consists of 2,300 patient records, each containing 15 input features (clinical, demographic, and diagnostic variables) and 1 output feature representing the target classification label. The input features include vital signs (heart rate, blood pressure, temperature), biochemical parameters, and medical history indicators.

In terms of demographics, the dataset covers patients aged 18 to 75 years, with a nearly balanced gender distribution (52% male, 48% female). Patient diversity is further reflected in varying health conditions, ranging from chronic diseases such as diabetes and hypertension to acute conditions requiring hospitalization. This heterogeneity improves the generalizability of the predictive model.

Preprocessing steps were applied to ensure data quality and consistency. Missing values were handled using mean imputation for numerical features and mode imputation for categorical variables. Outliers were detected through the Z-score method and corrected where clinically justified. All numerical variables were normalized using minmax scaling to eliminate magnitude bias, and categorical variables were converted into numerical form via Target encoding. Finally, the dataset was split into training (80%), and testing (20%) subsets. Table 2 and figure 9 present the detailed variable descriptions and their feature distributions, respectively.

Table 2. Attribute Details of the Real-Time Dataset

S.No	Attribute Name	Description
1	Gender	The gender of the patient
2	Age	Age of the patient
3	Blood Pressure	Systolic blood pressure level
4	Pulse Rate	Pulse rate level

5	Temperature	Body temperature
6	Respiratory Rate	Respiratory rate level
7	SpO_2	Oxygen saturation level
8	Complaints	Admitted with complaints
9	Obesity	1=Obesity; 0=No
10	Anemia	1=Anemia; 0=No
11	Cholesterol	Total; cholesterol level
12	Glucose	Fasting glucose level
13	Asthma	1= Asthma; 0=No
14	Hypertension	1= Hypertension; 0=No
15	Diabetes	1= Diabetes; 0=No
16	CVD	Output class. 1=Presence of cardiovascular disease; 0=No disease

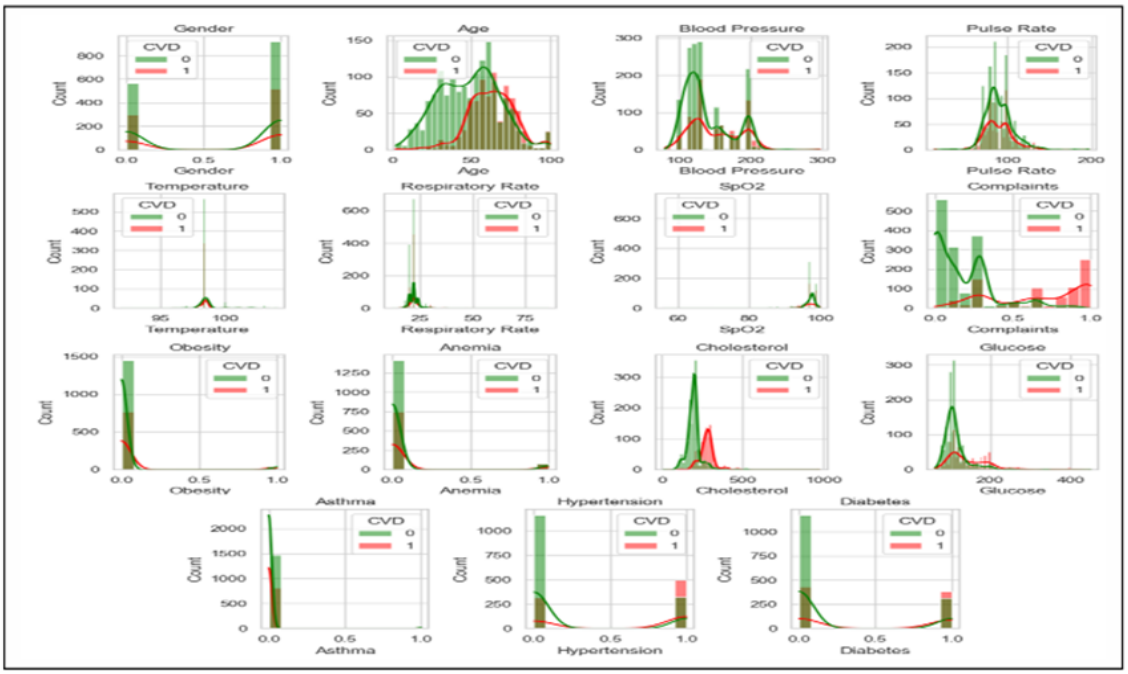

Figure 9. Feature Distribution of the Real-time Dataset

4.2 Benchmark Dataset Details

Table 3 shows the benchmark combined dataset's attribute details, which were retrieved from the UGC repository. This dataset comprises the Cleveland, Hungarian, Swiss, Long

Beach, VA, and Statlog Heart datasets. It contains 1,190 records, 11 input features and one output feature. Figure 10 demonstrates the benchmark dataset's feature distribution.

Table 3. Attribute Details of the Benchmark Dataset

S.No	Attribute Name	Description
1	Age	Age of the patient [Years]
2	Sex	Sex of the patient [1=Male; 0=Female]
3	Chest Pain Type	There are four different types of chest pain [0=Typical Angina; 1=Atypical Angina; 2=Non-Anginal pain; 3=Asymptomatic]
4	Resting EGC	Resting Electrocardiogram results [0=Normal; 1=ST-T Abnormality; 2=Probable LVH]
5	Resting BP	Resting blood pressure level (mm Hg)
6	Cholesterol	Serum cholesterol level (mg/dl)
7	Fasting BS	Fasting blood pressure [1: if Fasting BS > 120 (mg/dl), 0: otherwise]
8	Maximum Heart Rate	Maximum heart rate achieved during exercise
9	Oldpeak	ST depression from exercise
10	Exercise Angina	Agnosia was discovered after exercising [1=Yes; 0=No]
11	ST Slope	The slope of the peak exercise ST segment [1=Down sloping, 2=Flat; 3=Up sloping]
12	Heart Disease	Output class. 1=Presence of cardiovascular disease; 0=No disease

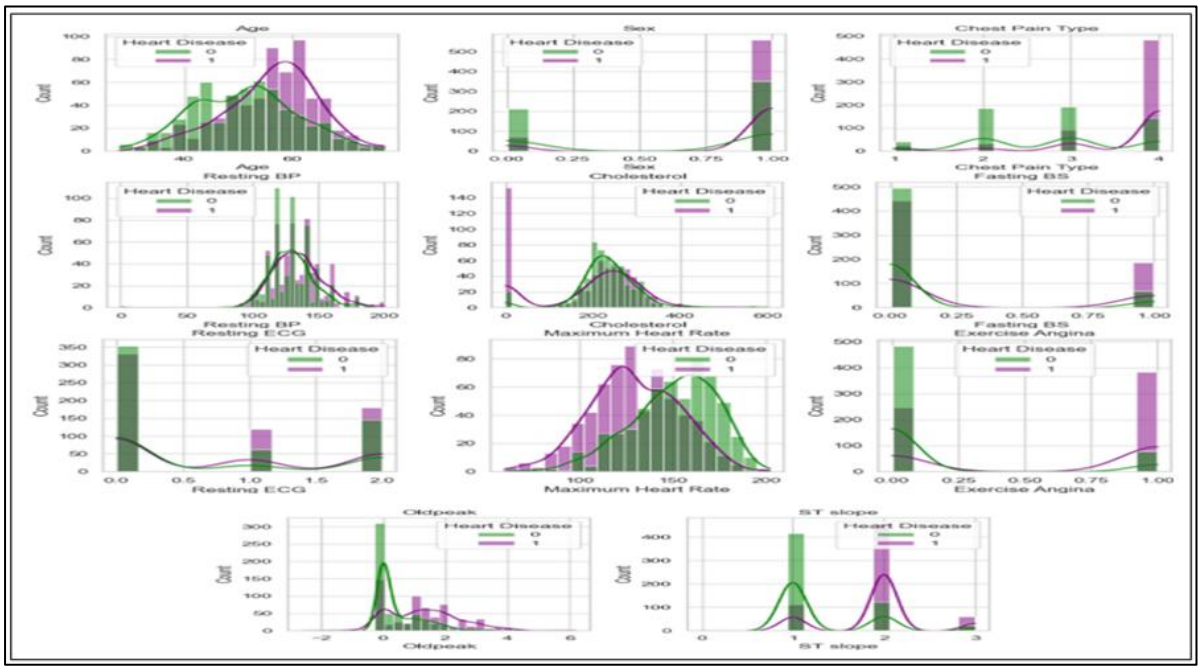

Figure 10. Features Distribution of the Benchmark Dataset

5. Implementation on the Real-Time Dataset

Figure 11 displays the results of the attribute selection methods on the real-time CVD dataset. For implementation, the best ten features were selected from each feature selection method. According to the Pearson correlation method, Cholesterol, Hypertension, Glucose, Age, Diabetes, Blood Pressure, Complaints, Obesity, and others are the first ten best features. According to the Recursive Feature Elimination method, Cholesterol, Age, Glucose, Complaints, Hypertension, Pulse Rate, Blood Pressure, SpO_2, Respiratory Rate, and Temperature were the first ten best features. In the Random Forest Feature Importance method, Cholesterol, Complaints, Hypertension, Blood Pressure, Age, Pulse Rate, Diabetes, SpO_2, Obesity, and Glucose are selected as the top ten features.

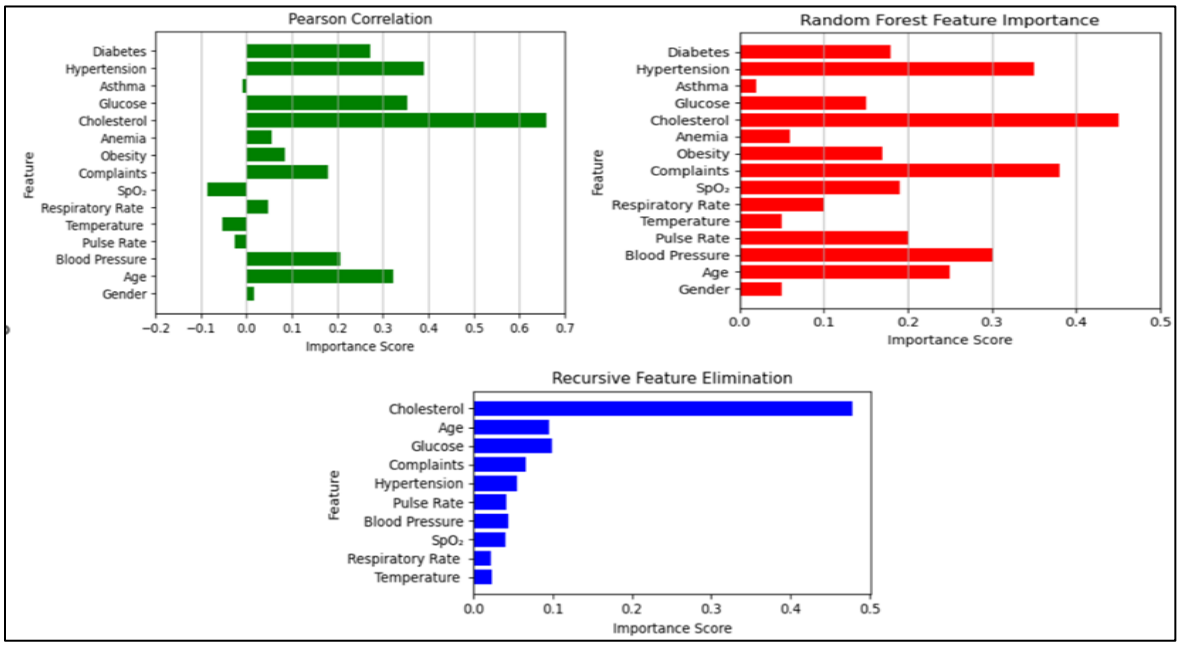

Figure 11. Attribute Selection Methods on the Real-time Data

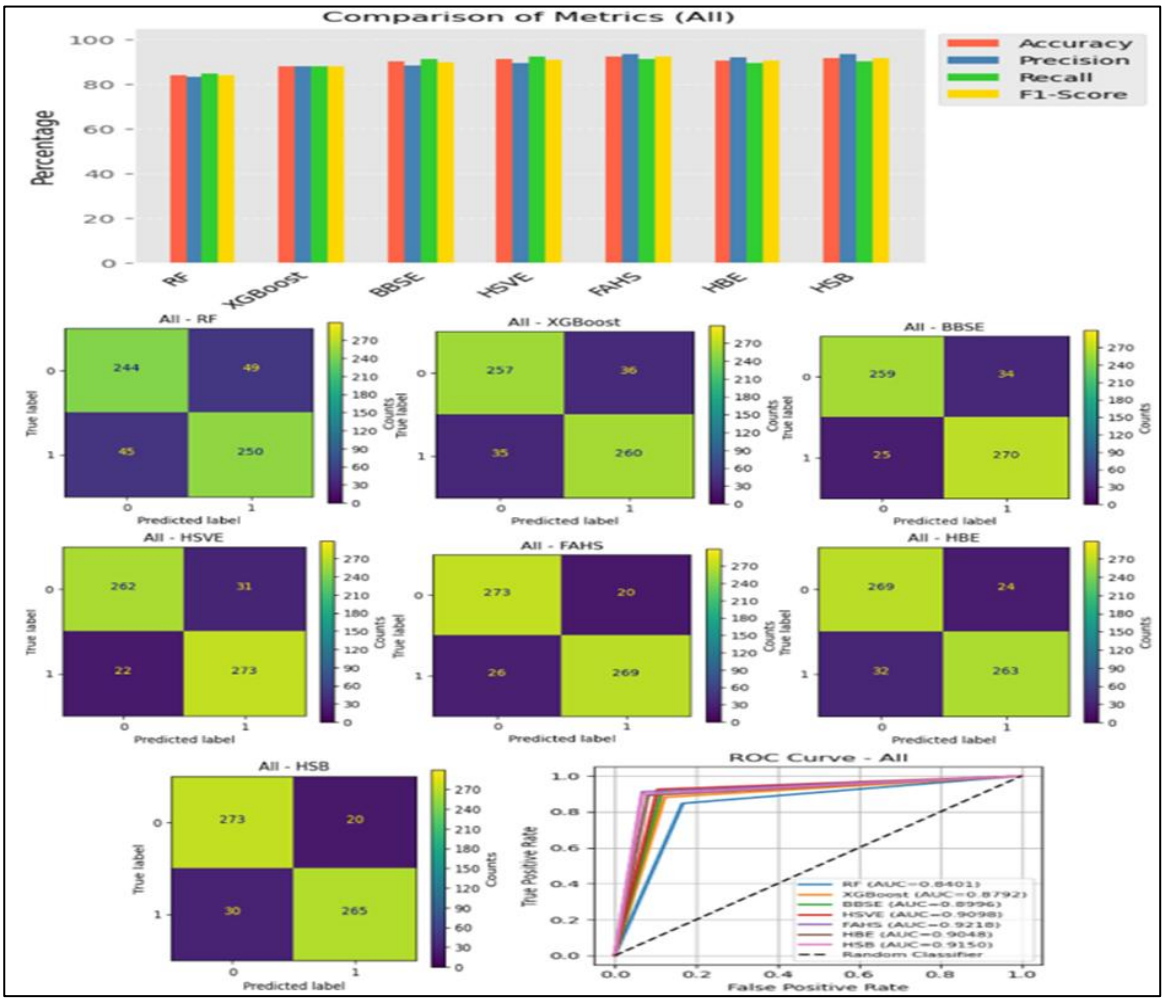

Figure 12. Performance Comparison Using All Features on the Real-time Dataset

Figure 12 shows performance measures, namely accuracy, recall, precision, F1 score, the confusion matrix, and the ROC curve when employing all real-time data elements. The FAHS model has the greatest accuracy score and AUC value out of all the classifiers. Figure 13 displays a variety of performance measures based on features chosen using the Pearson correlation coefficient method on real-time data. Among the classifiers, FAHS had the highest accuracy score and AUC value.

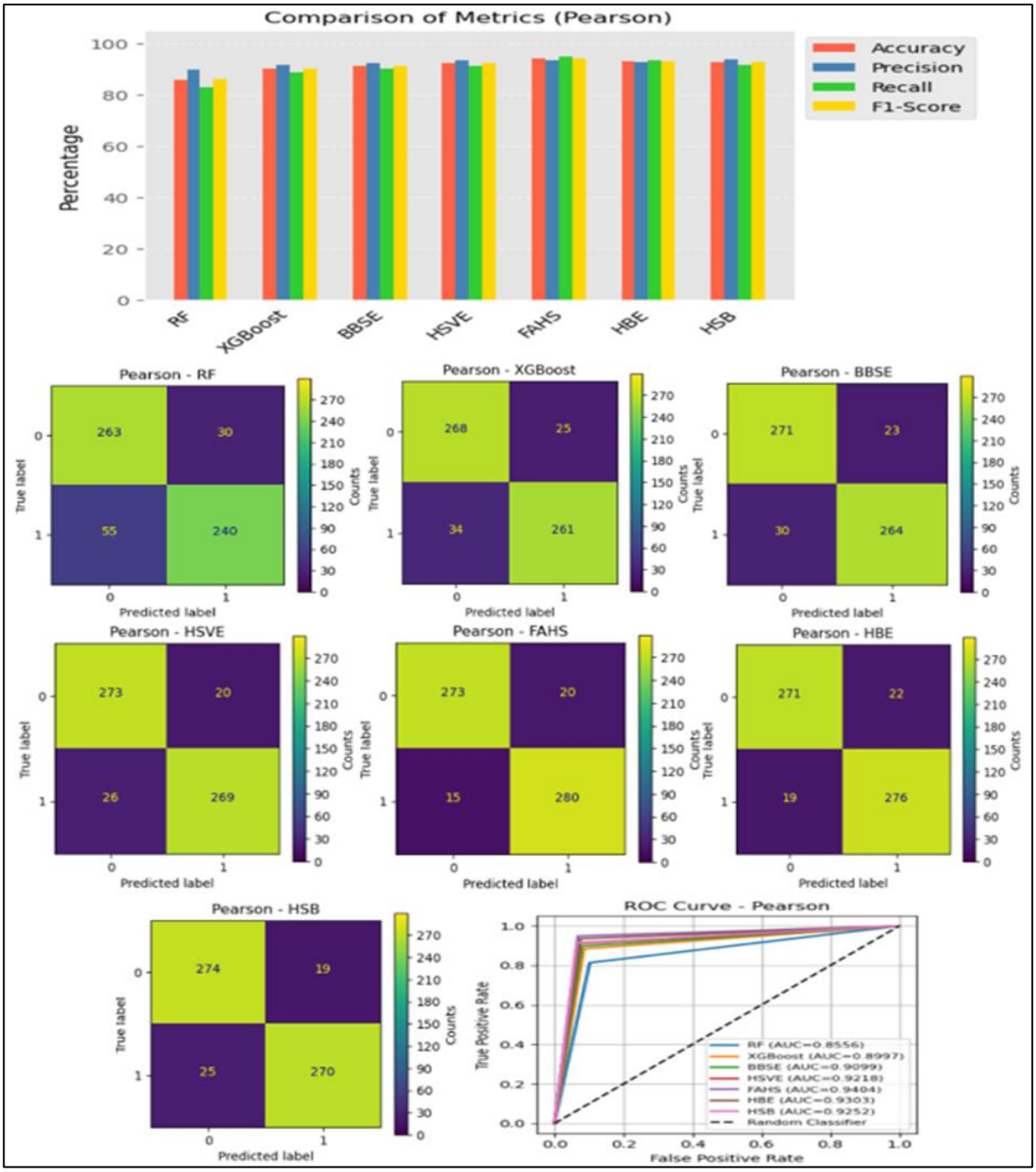

Figure 13. Performance Comparison Using Pearson Features on the Real-time Dataset

Figure 14 shows a range of performance metrics on a real-time dataset depending on the features selected using the recursive feature elimination method. The FAHS achieved the highest accuracy score and the highest AUC value.

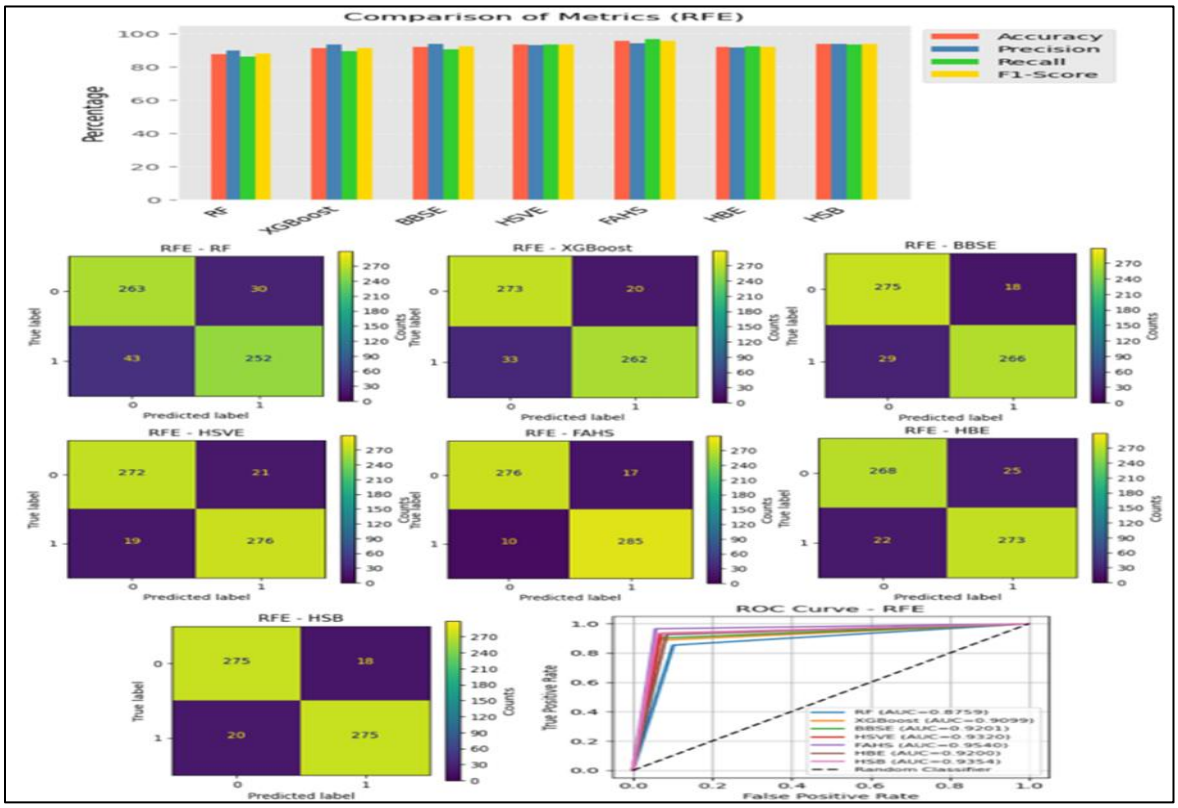

Figure 14. Performance Comparison using RFE Features on the Real-time Dataset

Using features, the RFFI chose based on real-time data, Figure 15 displays performance measures: accuracy, recall, precision, F1 score, the confusion matrix, and the ROC curve. The FAHS has the highest accuracy score and AUC value among all the classifiers.

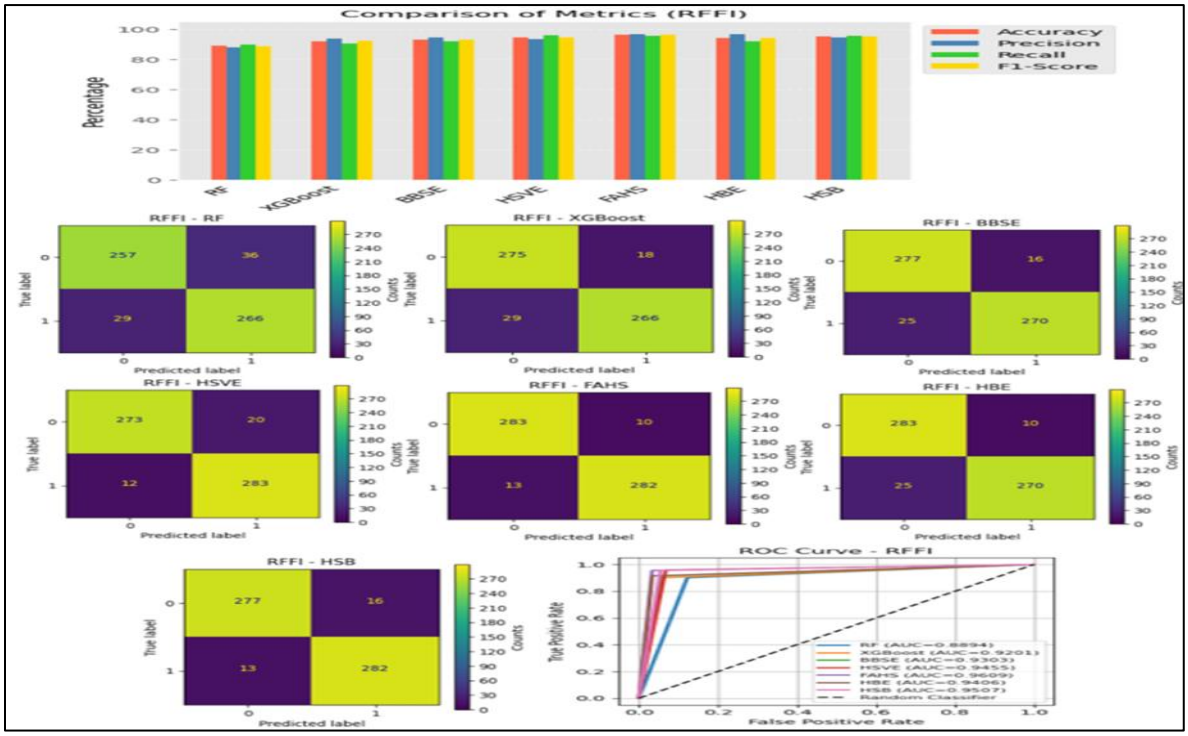

Figure 15. Performance Comparison Using RFFI Features on the Real-time Dataset

6. Model Performance Comparison

6.1 Model Performance Comparison on the Real-Time Dataset

Figure 16 shows a metric comparison of the real-time dataset, without feature selection and K-Fold CV, as well as with feature selection and K-Fold CV. Without feature selection and K-Fold CV, the FAHS had the best accuracy (92.18%). Then, feature selection and K-Fold CV were utilised for experimentation. While using the Pearson correlation coefficient feature selection, FAHS produced the highest accuracy of 94.05%. With the Recursive Feature Elimination feature selection method, FAHS scored the highest accuracy (95.41%). While employing the Random Forest Feature Importance technique, FAHS produced the best accuracy of 96.09%. The results section lacks essential statistical validation, including variance analysis and confidence intervals, which are critical for assessing the reliability and generalizability of the findings. Without these measures, it is difficult to determine the stability of model performance across different patient cohorts or datasets. Incorporating such statistical analyses would strengthen the credibility of the results, provide clearer insights into prediction uncertainty, and support more robust clinical interpretation.

Feature Selection	Model	Accuracy	Precision	Recall	F1 Score
All Features	RF	0.8401	0.8361	0.8475	0.8418
All Features	XGBoost	0.8793	0.8784	0.8814	0.8799
All Features	BBSE	0.8997	0.8882	0.9153	0.9015
All Features	HSVE	0.9099	0.898	0.9254	0.9115
All Features	FAHS	0.9218	0.9308	0.9119	0.9212
All Features	HBE	0.9048	0.9164	0.8915	0.9038
All Features	HSB	0.915	0.9298	0.8983	0.9138
Pearson	RF	0.8554	0.8889	0.8136	0.8496
Pearson	XGBoost	0.8997	0.9126	0.8847	0.8985
Pearson	BBSE	0.9099	0.9199	0.898	0.9088
Pearson	HSVE	0.9218	0.9308	0.9119	0.9212
Pearson	FAHS	0.9405	0.9333	0.9492	0.9412
Pearson	HBE	0.9303	0.9262	0.9356	0.9309
Pearson	HSB	0.9252	0.9343	0.9153	0.9247
RFE	RF	0.8759	0.8936	0.8542	0.8735
RFE	XGBoost	0.9099	0.9291	0.8881	0.9081
RFE	BBSE	0.9201	0.9366	0.9017	0.9188
RFE	HSVE	0.932	0.9293	0.9356	0.9324
RFE	FAHS	0.9541	0.9437	0.9661	0.9548
RFE	HBE	0.9201	0.9161	0.9254	0.9207
RFE	HSB	0.9354	0.9386	0.9322	0.9354
RFFI	RF	0.8895	0.8808	0.9017	0.8911
RFFI	XGBoost	0.9201	0.9366	0.9017	0.9188
RFFI	BBSE	0.9303	0.9441	0.9153	0.9294
RFFI	HSVE	0.9456	0.934	0.9593	0.9465
RFFI	FAHS	0.9609	0.9658	0.9559	0.9608
RFFI	HBE	0.9405	0.9643	0.9153	0.9391
RFFI	HSB	0.9507	0.9463	0.9559	0.9511

Figure 16. Metrics Comparison of the Real-time Dataset

While comparing all feature selection methods and classifiers on the real-time dataset, the random forest feature importance method and the FAHS combination had the highest accuracy of 96.09%.

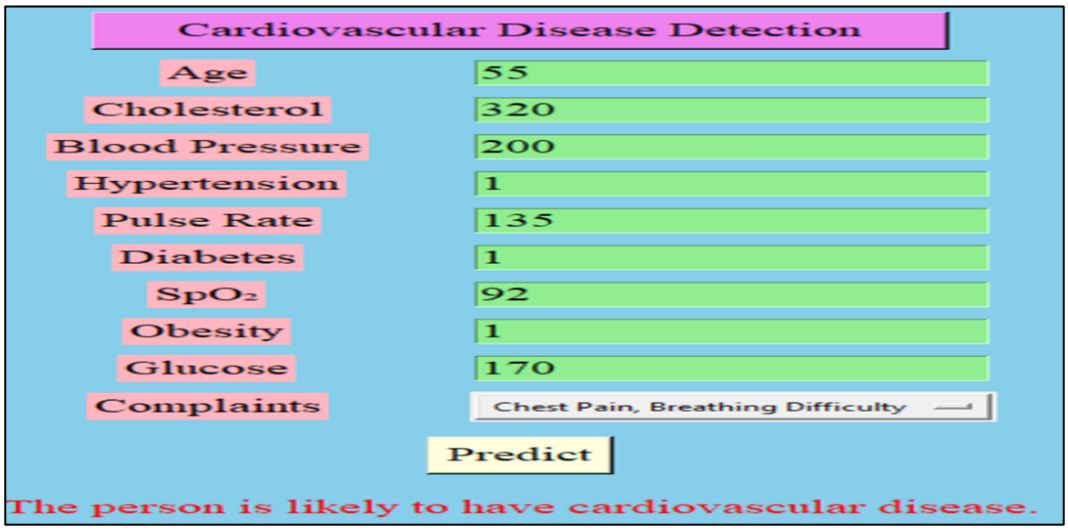

Figure 17. Predictive System on the Real-time Dataset

The cardiovascular disease detection system was created using the Python programming language. To provide an easy-to-use user interface, this application was developed as a desktop-based system using the built-in Python library Tkinter. Internet access is not necessary, and it is lightweight and simple to maintain. This predictive system uses the FAHS model with RFFI traits to detect CVD. Figure 19 shows that the patient with age, cholesterol, blood pressure, hypertension, pulse rate, diabetes, SpO_2, obesity, glucose, and complaints values of 55, 320, 200, 1, 135, 1, 92, 1, 170, and "chest pain, breathing difficulty," respectively, has cardiovascular disease. Model Performance Comparison on the Benchmark Data

Figure 18 shows a metric comparison of the benchmark dataset, without feature selection and K-Fold CV, as well as with feature selection and K-Fold CV. Without feature selection and K-Fold CV, the FAHS had the best accuracy (88.67%)

Feature Selection	Model	Accuracy	Precision	Recall	F1 Score
All Features	RF	0.8128	0.8155	0.8155	0.8155
All Features	XGBoost	0.8276	0.8333	0.8252	0.8293
All Features	BBSE	0.8424	0.8447	0.8447	0.8447
All Features	HSVE	0.8719	0.8738	0.8738	0.8738
All Features	FAHS	0.8867	0.8922	0.8835	0.8878
All Features	HBE	0.8621	0.8641	0.8641	0.8641
All Features	HSB	0.8818	0.8911	0.8738	0.8824
Pearson	RF	0.8276	0.8333	0.8252	0.8293
Pearson	XGBoost	0.8522	0.8544	0.8544	0.8544
Pearson	BBSE	0.8571	0.8627	0.8544	0.8585
Pearson	HSVE	0.9015	0.9029	0.9029	0.9029
Pearson	FAHS	0.9113	0.9126	0.9126	0.9126
Pearson	HBE	0.8818	0.8911	0.8738	0.8824
Pearson	HSB	0.9015	0.9029	0.9029	0.9029
RFE	RF	0.8424	0.8447	0.8447	0.8447
RFE	XGBoost	0.8571	0.8627	0.8544	0.8585
RFE	BBSE	0.8621	0.8641	0.8641	0.8641
RFE	HSVE	0.8916	0.8932	0.8932	0.8932
RFE	FAHS	0.9261	0.9314	0.9223	0.9268
RFE	HBE	0.9113	0.9126	0.9126	0.9126
RFE	HSB	0.9212	0.9223	0.9223	0.9223
RFFI	RF	0.8571	0.8627	0.8544	0.8585
RFFI	XGBoost	0.8719	0.8738	0.8738	0.8738
RFFI	BBSE	0.8818	0.8911	0.8738	0.8824
RFFI	HSVE	0.9212	0.9223	0.9223	0.9223
RFFI	FAHS	0.9409	0.9417	0.9417	0.9417
RFFI	HBE	0.9015	0.9109	0.8932	0.9020
RFFI	HSB	0.9261	0.9314	0.9223	0.9268

Figure 18. Metrics Comparison of the Benchmark Dataset

After applying feature selection and K-Fold CV, in the Pearson correlation coefficient feature selection, FAHS produced the highest accuracy of 91.13%. With the Recursive Feature Elimination feature selection method, FAHS scored the highest accuracy (92.61%). While employing the Random Forest Feature Importance technique, FAHS produced the best accuracy of 94.09% When comparing all feature selection methods and classifiers on the benchmark dataset, the random forest feature importance method and the FAHS combination had the highest accuracy of 94.09%.

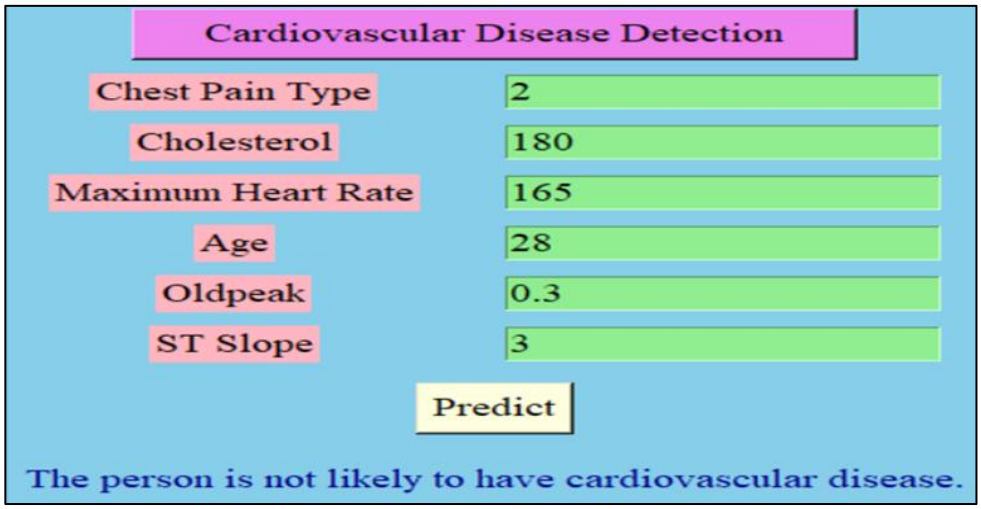

Figure 19. Predictive System on the Benchmark Dataset

Figure 19 shows that patients with Chest Pain Type, Cholesterol, Maximum Heart Rate, Age, Oldpeak, and ST Slope values of 2, 180, 165, 28, 0.3, and 3, respectively, do not have cardiovascular disease. This detection was made using the RFFI attributes and the FAHS model.

7. Comparative Analysis

Table 4 compares the performance of the suggested work with that of other studies. R. Aggrawal et al. [8] classified the data using sequential feature selection and random forest with an accuracy of 86.67%.

Table 4. Accuracy Comparison with Other Studies

Reference	Feature selection, K-Fold CV	Classifier	Maximum Accuracy%
R. Aggrawal et al. [8]	Sequential Feature Selection, k-fold	RR, LDA, SVM, KNN, DT, GBoost	86.67 (RF)
Spencer et al. [14]	Chi-squared, ReliefF, SU, PCA 10-fold CV	Bayes Net Classifier, KNN, DT, SVM, AdaBoost	85.0 (Chi with BN)
Takci [15]	ReliefF Method	NB, KNN, RF, SVM (Linear kernel)	84.81 (SVM Linear)
Zhaobin Qiu et al. [16]	LASSO	OCSCatBoost, LR, DT, KNN	73.67 (OCSCatBoos)

Snigdha Datta. [19]	Wald and Likelihood-Ratio, 10-fold CV	Logistic Regression	81.0
Annwesha Banerjee Majumde et al. [22]	10-fold CV	Bagged Ensemble (KNN, GNB, LR)	82.8 (LR) 82.5 (GNB) 83.2 (KNN)
Zhang et al. [23]	PCA	Stacking (Many Classifiers)	87.7
Tomar and Agarwal [24]	F-Score Feature Selection	LSTSVM	85.59
Dwi Normawati et al. [25]	VPRS, MFS 10-fold, 5-fold, 3-fold	Rule-based classifier derived using the VPRS approach	76.34 (VPRS with 5-fold CV)
Proposed work: Using Real-Time Dataset	Random Forest Feature Importance	BBSE HSVE FAHS HBE HSB	93.03 94.56 96.09 94.05 95.07
Proposed work: Using the Benchmark Dataset	Random Forest Feature Importance	BBSE HSVE FAHS HBE HSB	88.18 92.12 94.09 90.15 92.61

Spencer et al. [14] predict heart disease using the chi-squared feature selection method and a Bayes net classifier with 85.0% accuracy. Using the reliefF method and the SVM (linear kernel), Takci [15] showed 84.81% accuracy in HD prediction. Zhaobin Qiu et al. [16] predict heart disease using the LASSO feature selection method and an OCSCatBoost with 73.67% accuracy. Snigdha Datta [19] used 10-fold CV and LR to score 81% accuracy in CVD prediction. Annwesha Banerjee Majumde et al. [22] predict heart disease using 10-fold CV and a bagged ensemble with a maximum accuracy of 83.2% on KNN. Zhang et al. [23] classified the data using stacking, achieving an accuracy of 87.7%. Tomar and Agarwal [24] diagnosed HD using LSTSVM and F-Score Feature Selection with an accuracy of 85.9%. When Dwi Normawati et al. [25] employed VPRS and MFS with varying folds, the VPRS model produced a high accuracy of 76.34% using 5-fold cross-validation. When compared to previous experiments, the majority of the suggested models with different feature selection techniques showed improved accuracy (Figures 18 and 20). In Table 2, I emphasized the best feature selection technique and classifier combination that produced the greatest results.

8. Conclusion and Future Work

In this research, missing values are handled using a feature-type-based method, and the Z-score technique identifies outliers. Class imbalance is addressed using the SMOTE approach. Three feature selection algorithms, seven classification approaches, one cross-validation method, and several performance evaluation metrics were used for cardiovascular disease (CVD) detection. Experiments were run with and without a variable selection process to determine the impact of variable selection. The Pearson correlation coefficient, RFE, and RFFI methods were used as feature selection techniques. On the UCI cardiovascular disease datasets and a real-time dataset, several analytical approaches, including RF, XGBoost, BBSE, HSVE,

FAHS, HBE, and HSB, were used, and the results were compared with other studies. By testing the algorithm on several subsets of the data, the 5-fold cross-validation helps to minimize overfitting. If a model performs consistently well across all five folds, it is likely to be more robust and less prone to overfitting. According to feature selection algorithms, in real-time data, Cholesterol, Hypertension, Blood Pressure, and Age are the most essential and suitable features for classifying cardiovascular disease and healthy individuals. In the benchmark dataset, according to feature selection algorithms, ST slope, Oldpeak, Chest Pain Type, and Maximum Heart Rate are the most important and suitable features for classifying cardiovascular disease and healthy individuals. The FAHS model combined with the RFFI method achieved the highest accuracy of 96.09% on the real-time dataset and 94.09% on the benchmark dataset. When comparing both datasets, the real-time data achieved the highest accuracy. In this research, accuracy increased using feature selection, the k-fold cross-validation technique, and ensemble classifiers, and the purpose was accomplished as expected. This research offers a brand-new framework for detecting CVD diseases that consists of five creative ensemble-based methods. These methods use a number of sophisticated techniques, such as feature-augmented stacking, bootstrapped sampling, K-Fold cross-validation, sequential boosting, bagging and boosting integration, diverse classifiers, and soft voting ensemble approaches. While each method uses a different subset of these strategies, their combined use highlights the overall novelty and potency of the suggested framework.

Future models can be developed by assembling (hybrid) algorithms that combine several attribute selection methods to obtain the best attribute subsets. These suggested models could be expanded to address issues in industries other than healthcare, such as banking and agriculture.

References

[1] Trevisan, Caterina, Giuseppe Sergi, and Stefania Maggi. "Gender differences in brain-heart connection." In Brain and heart dynamics, Cham: Springer International Publishing, (2020): 937-951.

[2] M. S. Oh and M. H. Jeong, ‘‘Sex differences in cardiovascular disease risk factors among Korean adults,’’ Korean J. Med., vol. 95, no. 4, Aug. (2020): 266–275. http://doi.org/10.3904/kjm.2020.95.4.266

[3] World Health Organization, and J. Dostupno. "‘Cardiovascular diseases: Key facts." vol 13 (2016): 6.

[4] Uyar, Kaan, and Ahmet İlhan. "Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks." Procedia computer science 120 (2017): 588-593.

[5] Ayon, Safial Islam, Md Milon Islam, and Md Rahat Hossain. "Coronary artery heart disease prediction: a comparative study of computational intelligence techniques." IETE Journal of Research 68, no. 4 (2022): 2488-2507.

[6] Srivastava, Keshav, and Dilip Kumar Choubey. "Heart disease prediction using machine learning and data mining." International Journal of Recent Technology and Engineering 9, no. 1 (2020): 212-219.

[7] Ang, Jun Chin, Andri Mirzal, Habibollah Haron, and Haza Nuzly Abdull Hamed. "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection." IEEE/ACM transactions on computational biology and bioinformatics 13, no. 5 (2015): 971-989.

[8] Aggrawal, Ritu, and Saurabh Pal. "Sequential feature selection and machine learning algorithm-based patient's death events prediction and diagnosis in heart disease." SN Computer Science 1, no. 6 (2020): 344.

[9] Alam, Md Mahbub, Swapnil Saha, Proshib Saha, Fernaz Narin Nur, Nazmun Nessa Moon, Asif Karim, and Sami Azam. "D-care: A non-invasive glucose measuring technique for monitoring diabetes patients." In Proceedings of International Joint Conference on Computational Intelligence: IJCCI 2018, Singapore: Springer Nature Singapore, (2019): 443-453.

[10] Mienye, Ibomoiye Domor, Yanxia Sun, and Zenghui Wang. "An improved ensemble learning approach for the prediction of heart disease risk." Informatics in Medicine Unlocked 20 (2020): 100402.

[11] Wang, Haolin, Zhilin Huang, Danfeng Zhang, Johan Arief, Tiewei Lyu, and Jie Tian. "Integrating co-clustering and interpretable machine learning for the prediction of intravenous immunoglobulin resistance in kawasaki disease." Ieee Access 8 (2020): 97064-97071.

[12] Tama, Bayu Adhi, Sun Im, and Seungchul Lee. "Improving an intelligent detection system for coronary heart disease using a two-tier classifier ensemble." BioMed Research International 2020, no. 1 (2020): 9816142.

[13] Mishra, Jyoti, and Sandhya Tarar. "Chronic disease prediction using deep learning." In International Conference on Advances in Computing and Data Sciences, Singapore: Springer Singapore, (2020): 201-211.

[14] Spencer, Robinson, Fadi Thabtah, Neda Abdelhamid, and Michael Thompson. "Exploring feature selection and classification methods for predicting heart disease." Digital health 6 (2020): 2055207620914777.

[15] Takci, Hidayet. "Improvement of heart attack prediction by the feature selection methods." Turkish Journal of Electrical Engineering and Computer Sciences 26, no. 1 (2018): 1-10.

[16] Qiu, Zhaobin, Ying Qiao, Wanyuan Shi, and Xiaoqian Liu. "A robust framework for enhancing cardiovascular disease risk prediction using an optimized category boosting model." Mathematical Biosciences and Engineering 21, no. 2 (2024): 2943-2969.

[17] Pathan, Muhammad Salman, Avishek Nag, Muhammad Mohisn Pathan, and Soumyabrata Dev. "Analyzing the impact of feature selection on the accuracy of heart disease prediction." Healthcare Analytics 2 (2022): 100060.

[18] Osei-Nkwantabisa, Akua Sekyiwaa, and Redeemer Ntumy. "Classification and Prediction of Heart Diseases using Machine Learning Algorithms." arXiv preprint arXiv:2409.03697 (2024).

[19] Snigdha Datta, “Robust Cardiovascular Disease Prediction Using Logistic Regression”, The Journal of Management and Engineering Integration Vol. 14, No. 1 | Summer 2021

[20] Abdar, M. “Using Decision Trees in Data Mining for Predicting Factors Influencing of Heart Disease”, Carpathian Journal of Electronic and Computer Engineering, Volume 8, Issue 2, pages 31–36.

[21] Latha, C. Beulah Christalin, and S. Carolin Jeeva. "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques." Informatics in Medicine Unlocked 16 (2019): 100203.

[22] Majumder, Annwesha Banerjee, Somsubhra Gupta, and Dharmpal Singh. "An Ensemble Heart Disease Prediction Model Bagged with Logistic Regression, Naïve Bayes and K Nearest Neighbour." In Journal of Physics: Conference Series, vol. 2286, no. 1, p. 012017. IOP Publishing, 2022.

[23] Zhang, Jingyi, Huolan Zhu, Yongkai Chen, Chenguang Yang, Huimin Cheng, Yi Li, Wenxuan Zhong, and Fang Wang. "Ensemble machine learning approach for screening of coronary heart disease based on echocardiography and risk factors." BMC Medical Informatics and Decision Making 21, no. 1 (2021): 187.

[24] Tomar, Divya, and Sonali Agarwal. "Feature selection based least square twin support vector machine for diagnosis of heart disease." International Journal of Bio-Science and Bio-Technology 6, no. 2 (2014): 69-82.

[25] Dwi Normawati, Dewi Pramudi Ismi,” K-Fold Cross Validation for Selection of Cardiovascular Disease Diagnosis Features by Applying Rule-Based Data Mining”, Signal and Image Processing Letters Vol. 1, No. 2, July (2019): 22-32, ISSN 2714-6677, https://simple.ascee.org/