ORIGINAL RESEARCH ARTICLE
UHD JOURNAL OF SCIENCE AND TECHNOLOGY

Improving Cardiovascular Disease Prediction through Stratified Machine Learning Models and Combined Datasets

Tara Yousif Mawlood[1], Alla Ahmad Hassan[2], Rebwar Khalid Muhammed[3], Aso M. Aladdin[4]*, Tarik A. Rashid[5], Bryar A. Hassan[5]

[1]Department of IT, Computer Science Institute, Sulaimani Polytechnic University, Sulaymaniyah, Iraq, [2]Department of Database, Computer Science Institute, Sulaimani Polytechnic University, Sulaymaniyah, Iraq, [3]Department of Network, Computer Science Institute, Sulaimani Polytechnic University, Sulaymaniyah, Iraq, [4]Department of Computer Science, College of Science, Charmo University, Sulaymaniyah, Iraq, [5]Department of Computer Science and Engineering, School of Science and Engineering, University of Kurdistan Hewler, Erbil, Iraq

ABSTRACT

The global rise in cardiovascular disease (CVD) cases underscores the critical need for accurate and early diagnostic solutions. This study introduces a robust machine learning (ML) framework for predicting CVD risk by integrating two large, feature-identical datasets containing clinical and biological indicators along with patient history. Seven classification algorithms – logistic regression, random forest (RF), support vector machine (SVM), Gaussian naive Bayes (GNB), gradient boosting (GB), K-nearest neighbors, and decision tree (DT) – were employed. A stratified sampling strategy was used to ensure balanced class distribution, and model performance was further validated using k-fold cross-validation to enhance robustness and generalizability. The datasets, sourced from the UCI repository, were pre-processed and evaluated using metrics such as accuracy, precision, F1-score, log loss, and error rate, with performance further assessed using confusion matrices. Results revealed that ensemble models, particularly RF and DT, achieved optimal performance with 100% accuracy, while stratification significantly improved the outcomes of SVM, GNB, and GB. The integration of datasets, stratified sampling, and k-fold validation effectively enhanced model reliability while minimizing overfitting. These findings highlight the potential of ML to support early CVD diagnosis and lay the groundwork for future research on hybrid models and real-world clinical applications.

Index Terms: Cardiovascular Disease, Gradient Boosting, Heart Disease, K-Nearest Neighbors, Logistic Regression, Naive Bayes, Support Vector Machine.

1. INTRODUCTION

The cardiovascular system, also known as the circulatory system, is considered one of the most vital systems in the human body, along with the liver, lungs, and other essential organs [1], [2]. Cardiovascular disease (CVD) has become one of the most prevalent illnesses in nations worldwide today. It is often caused by low blood and oxygen levels in the circulatory system as well as blood vessel stenosis [3], [4]. Medical centers today have access to numerous datasets specifically focused on heart disease diagnoses. The evaluation of diseases and recognizing objects both make significant utilization of machine learning (ML) techniques [5]. Using ML algorithms to diagnose diseases, enormous quantities of medical data are converted into information

Access this article online

DOI:10.21928/uhdjst.v9n1y2025.pp149-168
E-ISSN: 2521-4217
P-ISSN: 2521-4209

Corresponding author's e-mail: Aso M. Aladdin, Department of Computer Science, College of Science, Charmo University, Sulaymaniyah, Iraq. E-mail: aso.aladdin@chu.edu.iq

Received: 12-02-2025 Accepted: 11-05-2025 Published: 03-06-2025

that can improve predicting and decision-making. To help healthcare providers by improving the precision and accuracy of making decisions over disease detection and diagnosis, ML research has gained importance in healthcare. A pair of its goals is the development of machine-based evaluation systems and disease prediction [6], [7]. A number of symptoms are able to identify CVD: Hypertension, chest pain, high blood pressure, cardiac arrest, etc. Many CVD types are present, each having a different variety of symptoms. Such as: (1) Chest pain, dyspnea, and throat pain are symptoms of cardiac disease in the blood vessels. (2) CVD caused by irregular heartbeats: discomfort, a slowing heartbeat, chest pain, etc. The most typical symptoms are discomfort, shortness of breath, chest pain, etc. The most typical symptoms include fainting, shortness of breath, and chest pain. CVD can be led on by pre-mature births, diabetes, high blood pressure, cigarette smoking, drug usage, and drinking alcohol. A fever, exhaustion, dry cough, and skin rashes are signs that the infection has occasionally spread to the inner membranes of the heart. Bacteria, viruses, and parasites are the causes of heart infections. Heart disease kinds include the following: Angina pectoris, congenital heart disease, Cardiac failure, cardiac illness, high blood pressure, plaque in the arteries, and a slower beat of the heart. Many automated techniques are available nowadays, including deep neural networks, algorithmic learning, and data mining [18]. Heart disease risk is commonly predicted based on various factors such as insulin resistance, CVD, high bad cholesterol levels, age, gender, smoking or drinking habits, obesity, heart rate, and chest pain, as demonstrated in numerous previous studies on different cases [9]. Leveraging medical information gathered from real-world cases, technology has become increasingly effective in predicting heart disease. With advancements in ML, a core component of artificial intelligence, these technologies have reached new heights. ML provides powerful tools for medical diagnosis and disease prediction, significantly enhancing the quality of healthcare services. By analyzing real-life medical data, these systems can more accurately detect whether an individual is at risk for heart disease. Addressing this issue requires robust accuracy in ensuring data security and maintaining confidentiality between patients and physicians. This can be achieved through the implementation of well-established security algorithms [10]. ML technology, considered a component of artificial intelligence, represents the highest point of technological advancement. Due to its ability to provide medical diagnostic tools for disease prediction, ML plays a significant role in enhancing the quality of health services [11]. Despite this, many metaheuristic algorithms have been employed to classify data and optimize this problem through evolutionary evaluation [12], [13], [14]. Furthermore, the supervised ML approach includes an algorithm for classification that can be utilized for prediction [15].

The aim of this study is to determine if cardiac questions can be identified based on a patient's medical factors, such as age, gender, and chest pain. For this purpose, patient characteristics and medical selected data datasets as specified in the future. By analyzing this dataset, the goal is to determine whether a patient has a cardiac issue.

The novelty of this study lies in its innovative approach to enhancing CVD prediction by combining two datasets (UCI and HD) and applying stratified ML techniques. A key aspect of this novelty is the integration of two large datasets with identical feature variables, creating a more diverse and comprehensive dataset that improves model generalization compared to using a single source. In addition, the study employs stratified data splitting to maintain balanced class distributions, which is particularly beneficial for imbalanced datasets. This technique not only reduces model bias but also significantly improves the prediction accuracy of algorithms such as logistic regression (LR), random forest (RF), support vector machine (SVM), Gaussian Naive Bayes (GNB), gradient boosting (GB), K-nearest neighbors (KNN), decision tree (DT). The study further distinguishes itself through a thorough comparative performance evaluation of multiple ML algorithms – including LR, RF, SVM, GNB, GB, KNN, and DT – assessing their effectiveness with and without stratification. Notably, the research provides a quantitative analysis of how stratification positively impacts model accuracy, precision, and F1-score, underscoring its critical role in healthcare-related ML tasks. While the study leverages widely used ML methods, its uniqueness is evident in the dataset integration, the strategic use of stratification, and the detailed assessment of its effects on model performance. To further enhance its contributions, future work could explore advanced techniques such as deep learning (DL), hybrid models, or clinical validation using real-world healthcare data. This study's primary contribution lies in training health-related features using seven methods:

- Evaluate the performance of various ML algorithms, including LR, RF, GNB, GB, KNN, SVM, and DT, in predicting CVD.
- The combined dataset showed enhanced accuracy by using the stratify parameter, which ensured balanced training and improved model performance.
- This strategy helps medical professionals assess patient risk, showcasing ML's potential to enhance diagnostic accuracy and improve patient outcomes.

- The use of stratified techniques greatly enhanced the accuracy of SVM, GNB, and GB models, emphasizing the value of balanced data distribution during training.
- This study evaluates the performance of various classification algorithms on three distinct datasets: UCI, HD, and the combined UCI-HD dataset.
- The results highlight the effectiveness of RF and DT algorithms, demonstrating their robustness across all datasets, with RF achieving perfect accuracy on the combined dataset. In addition, the performance of other algorithms, such as KNN and GNB, provides valuable insights into their strengths, particularly on the HD dataset.
- This comparative analysis supports the selection of the most suitable algorithm for classification tasks across different datasets.

The rest of the paper is organized as follows: Section 2 delivers a background review, followed by Section 3, which outlines the methods and materials used. Section 4 presents the results and analysis, while Section 5 discusses the performance of the algorithms based on the adjustments. Finally, the conclusion and future work are presented in the past section. Procedure for Paper Submission.

2. BACKGROUND REVIEW

Many studies involving animals or humans, and other studies that require ethical approval, must list the authority that provided approval and the corresponding ethical approval code. Many studies are being carried out having the goal of using algorithms based on ML to identify cardiac disease. The study employed various ML methods to create a prediction model for categorizing cardiac diseases. The following part discusses a few of the earlier studies on predicting the likelihood of heart disease.

The DT, RF, and Naïve Bayes (NB) methods are applied to the Cleveland heart disease dataset by Gavhane *et al.* [16]. The study dataset was used to evaluate the accuracy of predictions of the approaches, and the RF strategy performed better than the DT and NB procedures. Ambekar and Phalnikar [17] use a heart disease dataset to compare the prediction power of many ML methods, such as GNB, LR, RF, and KNN. In terms of prediction accuracy, LR performed better than all other approaches according to the two different findings. The comparison of three ML algorithms, DT, RF, and multi-layer perception, using the Wisconsin Heart Disease Data Repository is carried out by Jothi *et al.* [18]. The methods are compared for accuracy in predicting cardiac illness, and the findings indicate that multi-layer perception and neural networks perform better in this regard. The Wisconsin Heart Disease Data Repository forecasts heart disease using NB, as described by Segie *et al.* [19]. It can achieve 87% overall prediction accuracy. In artificial ML and forecasting systems, the NB model performs better than the other models in terms of performance and the ability to accurately forecast cardiovascular illness, with an adequate forecasting accuracy of eighty-seven. The study conducted by Kajal and Nishika [20] focused on different methods of classification used to predict an individual's danger degree based on blood pressure, cholesterol levels, cardiac rate, age, gender, and other characteristics.

Data mining methods, including NB, KNN, DT, and Neural Network, are used to increase the accuracy of the hazard level. The kernel nearest neighbor and ID3 algorithms were used to determine the risk rate of heart disease. The accuracy rating for various amounts of attributes was also provided. Babu *et al.* [21], conducted research on a range of educational apps that support the identification of many cardiac conditions. A selection of methods, including data mining, SVMs, computationally intelligent classifiers, and hidden markov's models, were employed. Since treating heart disease is extremely costly and out of reach for the average person, these kinds of cutting-edge technologies are created to address this issue. The beginning predictions can also benefit from these strategies. It modifies daily routines somewhat to prevent more suffering. As a result, the author draws the conclusion that the anticipated strategy is highly helpful and offers several advantages. Kannan and Vasanthi [22] employed a variety of ML algorithms, including LR, RF, SVM, and stochastic gradient boosting, to identify potential cardiovascular illnesses. The equation predicts that LR has the most accuracy, coming in at 86.5%.

In addition, Raza [23] employed LR, NB, a multilayer perceptron, and a combined learning model to classify heart diseases. The result shows that combined learning has improved the prediction performance of cardiac disease when compared to other approaches. For example, Z-Alizadesh Sani and the Cleveland heart disorder dataset were two separate datasets utilized by Sapra *et al.* [24] to diagnose cardiac illnesses. These datasets were previously analyzed using six different ML techniques: LR, DL, DT, RF, SVM, and collaborative learning gradient boosted tree. When compared to other techniques, gradient boosted tree strategy yielded the highest accuracy, at 84%. Because CVD has one of the highest death rates worldwide, projection is an essential part of medicine. Several techniques involving ML are being used to effectively forecast cardiac disease. Research has demonstrated that technologies such as KNN, RF, DT, and SVMs can reliably forecast CVD with excellent accuracy rates of 79%–94% [25]. According to this research [26], NB (83.60%), KNN (90.16%), LR (86.88%), RF (96.72%), extreme gradient boost (95.08%), and DT (77.049%) were the ML techniques that were employed. Remarkably, the method known as RF performed more effectively than the remaining algorithms, with a maximum accuracy rate of 96.72%. For further clarification on the methods used, Table 1 includes a summary of the approaches, along with a comparison to previous works that have addressed similar topics.

3. MATERIALS AND METHODS

3.1. Dataset

The UCI Heart Disease dataset is a widely used dataset for predicting CVD, containing various attributes related to patient health, such as age, cholesterol levels, and electrocardiographic results. The investigation utilized two datasets for analysis, as summarized in Table 2. The first dataset, UCI, contains samples categorized based on the presence or absence of CVD. Similarly, the second dataset, HD, provides additional samples with the same classification criteria.

The Combined dataset, created by merging UCI and HD, offers a comprehensive view of CVD distribution. All

TABLE 1: Relevant bibliography

Title of research	The methods	Different types of data used	Year	References
Utilizing Artificial Intelligence to Predict Cardiovascular Disease	RF, DT, and NB	Cleveland Cardiovascular Disease Data Set	2018	[16]
Making use of convolutional neural networks to predict the probability of diseases	GNB, RF, KNN, and L R	Cardiovascular Disease Data Set	2018	[17]
Genetic algorithm-based feature determination technique in the healthcare dataset	RF, DT, and multi-layer neural networks for perception	Wisconsin Cardiovascular Research Center	2019	[18]
Support Vector Techniques for Machine-Based Detection of Cardiovascular Disease	NB	Wisconsin Cardiovascular Research Center	2019	[19]
Cardiovascular Disease Detection Utilizing Data Mining Methods	Neural Network, KNN, DT, and NB	Features: Sexuality, Age, Blood Pressure, Cholesterol Levels, and Heart Rate	2016	[20]
Diagnosing Cardiovascular Disease Through Data Mining Method	Data mining, computerized smart classifiers, SVM, and concealing Markov chain	Cardiovascular Disease Data Set	2017	[21]
ROC curve-based machine learning techniques for cardiovascular disease diagnosis and prognosis	Probabilistic Gradient Boosting, RF, SVM, and LR	Cardiovascular Disease Data Set	2019	[22]
Improving the identification of cardiovascular disease accuracy using a majority vote and group learning	LR, NB, Multilayer Perceptron, Ensemble Learning	Cardiovascular Disease Data Set	2019	[23]
Using a combined technique, a smart approach for detecting CAD	Gradient Boosted Tree, RF, SVM, DT, DL, and LR	Cleveland Cardiovascular Disease and Z- Alizadesh Sani Datasets	2021	[24]
Using Machine Learning Techniques to Track Cardiovascular Issues in Cardiovascular Disease	SVM, RF, DT, and KNN	Cardiovascular Disease Data Set	2023	[25]
An Organized Analysis of Machine Learning Algorithms for Heart Failure Prediction	KNN, LR, and RF	Cardiovascular Disease Data Set	2022	[25]

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, KNN: K-nearest neighbors, DT: Decision tree, DL: deep learning

TABLE 2: Cardiovascular disease dataset distribution and sources

Dataset	Feature	Samples	Negative samples (%)	Positive samples (%)	Source
UCI	14	1025	499 (48.68)	526 (51.32)	https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data
HD	14	303	138 (45.5)	165 (54.5)	https://www.kaggle.com/code/mragpavank/heart-disease-uci/notebook
UCI-HD	14	1328	637 (48.0)	691 (52.0)	Includes UCI -HD

datasets are complete, with no missing feature values. Fig. 1 illustrates the distribution of CVD across the samples. For model development, 80% of the combined dataset was allocated for training, and 20% for testing.

3.2. Data Pre-Processing

This study focuses on pre-processing the UCI and HD (heart disease) datasets before developing a predictive model using ML. These datasets have undergone extensive cleaning and pre-processing, making them easier to use and requiring minimal effort for data preparation. In addition, they are well-documented and frequently cited in scientific research. In both datasets, the target attribute is an integer indicating the presence of heart disease in a patient. A value of 0 signifies no heart disease, while a value of 1 indicates its presence. The attribute sex represents gender, with two classes: 1 for males and 0 for females. The attribute cp (chest pain type) includes four classes, while fbs (fasting blood sugar) has two classes. Similarly, restecg (resting electrocardiogram) consists of three classes, and exang (exercise-induced angina) has two classes. The attribute slope (ST slope) comprises three classes. Four additional attributes – trestbps (resting blood pressure), chol (cholesterol level), age, and oldpeak – are treated as numerical values. The data pre-processing process involves multiple steps, from data loading to splitting for training and testing. These steps are detailed in Table 3.

3.3. Selection Algorithms

There are several algorithms commonly used for the classification of cardiac disease, including LR, RF, SVM, GNB, GB, KNN, and DT.

3.3.1. LR algorithm

It consists of a classification process that makes use of only one class-based classifier and a single multinomial LR approach. When using a specific technique, a LR analysis

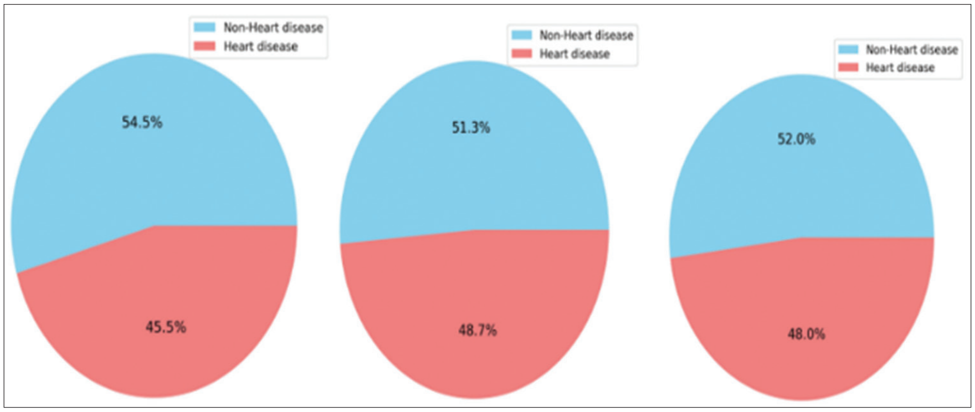


Fig. 1. Patient with cardiovascular disease compared to non-heart disorder patients.

TABLE 3: Heart dataset features

Attributes	Information
Sex	Gender of participants
Cp	Kind of chest pain. There are four parts to this feature: unusual angina, usual angina, and both. Pain that is not anginal and asymptomatic
Trestbps	Blood pressure of the patient during a time of rest or inactivity
Chol	Level of cholesterol
Fbs	The level of blood sugar has an accurate level if it is more than 120 mg/dL and an incorrect value if it is below 120 mg/dL.
Restecg	Electrocardiogram outcomes acquired as a patient is at rest, often known as resting electrocardiogram results. Following the Estes criteria, a result of 0 denotes normalcy, an amount of 1 suggests aberrant ST-T waves, & a result of 2 suggests a certain risk of hypertrophy of the left ventricle.
Thalach	Maximal heartbeat
Exang	The patient's pain whereas physical activity. Value, true if the answer is "yes" and false if the word "no"
Oldpeak	The reduction in ST brought on by activity.
Slope	Slope during exercise at maximum ST. The slope of it can be classified as upsloping, round, or downsloping.
Ca	The amount of vessels is identified by coloring.
Thal	Test for thalassemia has a total of three numbers: changeable error, immovable defect, and regular.
Target	Cardiac illness (1) is a type of label. Non-cardiac illness (0)
Thalach	Maximal heartbeat

usually shows where the group's borders are and how far removed the category's probabilities are from them. This approaches the extremes (0 and 1) more quickly depending on the data collection. LR is elevated beyond the level of a basic classification by these conditional arguments. It might be applied in a different method more offers more accurate and thorough estimates, although they are not certainties. LR is an estimating technique that resembles ordinary least squares regression. However, employing LR for prediction results in a single response [27]. LR has become one of the increasingly popular techniques for intermittent information analysis and statistical applications. The LR approach makes use of the linear interpolation method [28]. The sigmoid function, given by the equation (1)

$$sig(t) = \frac{1}{1+e^{-t}} \tag{1}$$

3.3.2. RF algorithm

Among the most effective learning techniques is RF. Humanities researchers are able to benefit from algorithmic advancements if they are allowed access to an application of the method. Since models that utilize trees are the foundation of the RF technique, let's start by talking about them. With a tree-based approach, the collection of data is repeatedly divided into two separate categories according to a pre-defined parameter or another is satisfied. The referred to as nodes of leaves, or leaves are located at the base of the DT. Comparing to DTs, the RF algorithm calculates the error rate more accurately. Particularly, it can be demonstrated analytically that the error rate constantly converges as the number of trees rises [29]. At the preparation phase, the out-of-bag (OOB) error approximates the variance of the RF mistake. A distinct bootstrapping sample serves as the foundation for every tree. Approximately one-third of the observations are randomly excluded from each bootstrapping dataset. An OOB example of this is the collection of each of this excluded information for an individual tree. When choosing an algorithm and fine-tuning factors, determining which ones will result in a small OOB variance is frequently crucial. Keep in mind that the size of the group of predictor factors is essential for regulating the trees' ultimate depth in the RF method. As a result, it is a parameter that must be adjusted when choosing a model.

3.3.3. SVM algorithm

The most commonly implemented supervised ML technique, SVM, is utilized for classification as well as regression. However, this approach is mostly examined for ML issues with classification. To swiftly place the newly acquired information in the right group, the strategy known as SVM aims to construct the most effective border, the line of sight that may divide the space with n dimensions into classes. This optimal selection of boundaries is referred to as a "hyperplane" [30]. SVM selects the extreme vectors that help to create the hyperspace. The highest and lowest vectors are collectively referred to as vectors of support, and the method that employs a disproportionate number of verticals is called SVM. The SVM image below classifies two distinct groups using hyperplanes or borders of selections.

3.3.4. KNN algorithm

The KNN algorithm, the most basic categorizing technique, is based on supervised learning techniques. The KNN technique can be applied to reversion, although it is primarily used for classification [22]. A new data point is categorized using the KNN algorithm according to how well the information it provides matches the existing data. How fresh data can be promptly classified by the method known as KNN when it falls into an appropriate class.

3.3.5. GNB algorithm

The GNB, which comes through the Bayesian theory, GNB provides constant data that are drawn through the Gaussian typical distribution. The idea that its elements are autonomous is the foundation of the GNB. This sorting algorithm is regarded as being one of the most straightforward and practical methods. For categorization using supervised ML, which is predicated on the concept that the information is regularly generated. GNB is demonstrated [31]. If every chosen feature contributes equally and independently, the Bayes theorem depends on multiplying the probability and previous by the evidence presented. Since they are independent of one another, it is assumed that the significance of each attribute has an equal impact on the result. The likelihood of a specific event because something else has previously occurred is known to be called the GNB using the formula (2).

$$(A|\,B) = \frac{P(B|A)\ P(A)}{P(\ B)} \tag{2}$$

This formula determines the following probability, or the likelihood that A will occur, considering that B will occur, and the likelihood that B will occur, provided that A will occur. The probabilities of A Occurring P(A) split by proof, or the possibility of B occurring P(B), is P(B | A), which is the probability multiply by the previous event.

3.3.6. GB

In terms of accuracy, the GB provides a cutting edge, particularly for supervised training tasks on structured datasets. Freund and Schapire created AdaBoost, the initial boosting method, in the year 1997. Combining various models of ML is the aim of collaborative learning with the objective of enhancing the accuracy of predictions. The precise concept behind boosting is that you should start with an algorithm that is referred to as a poor learner, meaning it is just marginally higher quality than chance guesswork. Over time, the algorithm gets better by fixing the mistakes made by the prior model at each stage. GB begins with a poor learner, usually a DT, and continuously refines this starting learner by accounting for the inaccuracy of the preceding models at each stage. GB has become considered among the most successful algorithmic learning strategies available. The most common form of boosting progressively advertises a single DT as time by using trees of choices. Higher accuracy is the result of the model's effectiveness being gradually improved by this sequential change. There are many kinds of GB available. The gradual boosting version used in this investigation relies on [32].

3.3.7. DT algorithm

The DT Algorithm is a supervised learning method for classification and regression. It splits data into subsets based on attribute values, forming a tree-like structure of decisions and outcomes. Metrics, such as entropy and information gain [33]. The entropy value is calculated using the formula in Equation 3, when S is representing a specific set, A is one of the attributes, the number of divisions for attributes is n, (|Si|) is the quantity of instance in partition (i), finally |S| is the sum of the instance in (S).

$$\text{Gain}(\text{S},\ \text{A}) = \sum_{i=1}^{n} \frac{|S1|}{S}\ \text{entropy}(\text{Si}) \tag{3}$$

3.4. Proposed Method

The proposed method, illustrated in Fig. 2, flowchart for the Proposed System, outlines a comprehensive approach to heart disease prediction using ML. The process begins with

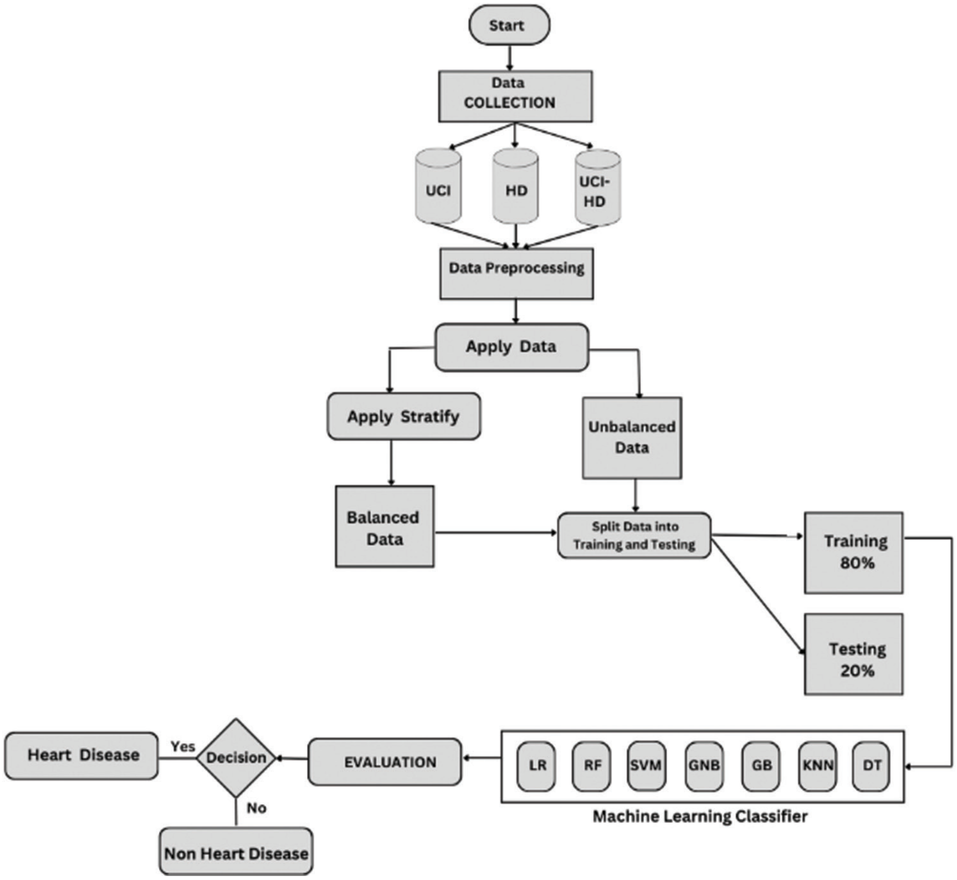


Fig. 2. Flowchart for the proposed system.

TABLE 4: Performance evaluation of machine learning algorithms on the UCI dataset

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	79.51219	0.80234	0.7937867	0.45313	0.2048780
RF	98.53658	0.98578	0.9853637	0.06658	0.0146341
SVM	88.78048	0.89226	0.8874512	0.25314	0.1121951
GNB	80.00000	0.81050	0.7981741	0.66268	0.2000000
GB	93.17073	0.93235	0.9316748	0.19782	0.0682926
KNN	83.41463	0.83869	0.8335281	0.25111	0.1658536
DT	98.5365	0.98578	0.985363	0.5274	0.014634

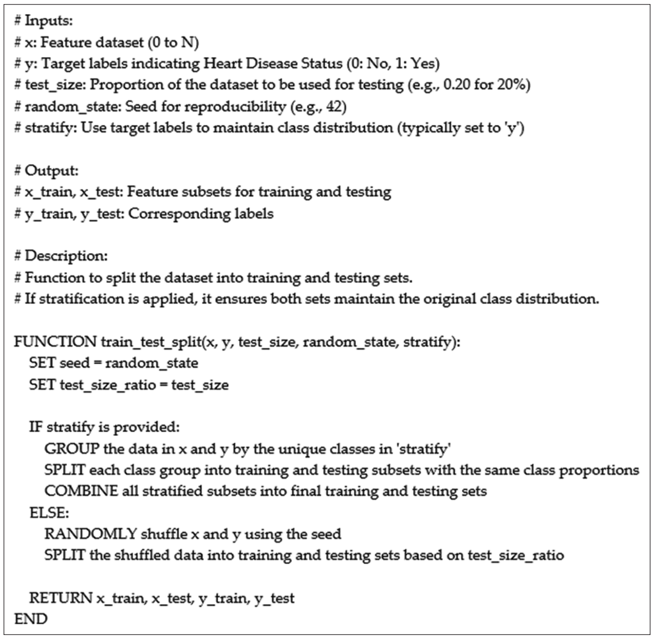
LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

data collection from three sources: UCI, HD, and UCI-HD datasets, ensuring a diverse and robust dataset. This data undergoes pre-processing to clean and normalize it, addressing issues, such as missing values, noise, and inconsistencies to prepare it for analysis. The next step involves applying the data. For balanced datasets, a stratification process is applied to ensure even class distribution, reducing bias. For unbalanced datasets, the data are split into training (80%) and testing (20%) subsets, enabling effective model evaluation. The processed data are fed into a ML classification pipeline consisting of algorithms, such as LR, RF, SVM, GNB, GB, KNN, and DT. These classifiers are trained on the dataset to build predictive models. Finally, the system undergoes an evaluation phase, classifying outcomes into two categories: Heart Disease and Non-Heart Disease, based on the decision boundary of the classifiers. This robust framework ensures accurate and fair predictions, with balanced data enhancing model reliability and unbalanced data reflecting real-world scenarios.

3.4.1. The importance of stratified sampling in dataset splitting

Table 4 shows that stratify = y is used in train_test_split, it ensures that the class distribution in the target labels (y) is maintained across both training and test sets. This is particularly useful for imbalanced datasets, where some classes may have fewer samples than others. The function groups the data by unique classes in y and splits each class proportionally into training and testing subsets. This process ensures that the relative frequency of each class in y remains consistent. If stratify is not specified, the split is purely random and may lead to class imbalances in the subsets.

The data processing in this study employs a stratified splitting method to ensure that the class distribution of the target labels is preserved across both training and testing subsets. The dataset is split into two parts: A training set and a testing set, with the proportion for the testing set specified by the test_size parameter (20%). The random_state seed ensures that the data split is reproducible. If the stratify option is enabled, the dataset is grouped by unique target label classes, and each group is split into training and testing sets while maintaining the original class proportions. These stratified splits are then combined into the final training and testing sets. If stratification is not applied, the function will randomly shuffle and split the data based on the test_size ratio. This stratified methodology ensures that the subsets are representative of the overall dataset, preserving the balance of target labels in both training and testing sets, which is critical for training accurate ML models. Fig. 3 illustrates the stratified dataset splitting for training and testing.



```
# Inputs:
# x: Feature dataset (0 to N)
# y: Target labels indicating Heart Disease Status (0: No, 1: Yes)
# test_size: Proportion of the dataset to be used for testing (e.g., 0.20 for 20%)
# random_state: Seed for reproducibility (e.g., 42)
# stratify: Use target labels to maintain class distribution (typically set to 'y')

# Output:
# x_train, x_test: Feature subsets for training and testing
# y_train, y_test: Corresponding labels

# Description:
# Function to split the dataset into training and testing sets.
# If stratification is applied, it ensures both sets maintain the original class distribution.

FUNCTION train_test_split(x, y, test_size, random_state, stratify):
  SET seed = random_state
  SET test_size_ratio = test_size

  IF stratify is provided:
    GROUP the data in x and y by the unique classes in 'stratify'
    SPLIT each class group into training and testing subsets with the same class proportions
    COMBINE all stratified subsets into final training and testing sets
  ELSE:
    RANDOMLY shuffle x and y using the seed
    SPLIT the shuffled data into training and testing sets based on test_size_ratio

  RETURN x_train, x_test, y_train, y_test
END
```

Fig. 3. Stratified dataset splitting for training and testing.

4. RESULTS

According to the methodology of this study, the datasets have been compared based on the comparative parameters for each algorithm described in subsection one. Notably, the results for each dataset are presented and the modifications evaluated are illustrated in subsection two.

4.1. Comparative Parameters

The performance of the proposed model was evaluated using the UCI, HD, and Combined (UCI-HD) datasets, which provide comprehensive information on CVD. After dividing the data, the model was trained and tested using algorithms such as LR, RF, GNB, GB, SVM, DT, and KNN. The algorithm with the highest efficiency was identified by analyzing performance metrics, including accuracy, precision, F1-score, and logarithmic loss. Accuracy, which measures the percentage of correctly classified samples, was calculated using the formula derived from the confusion matrix. This formula, referred to as equation 4, quantifies the model's ability to classify CVD cases accurately across different datasets.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Precision evaluates the accuracy of a classifier by comparing the number of true positives (TP) in the actual data to the number of predicted TP. This measure of accuracy is essential for assessing the performance of the proposed method, as calculated mathematically according to equation 5.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

F1-Score: The F1-score is a statistical metric used to evaluate the performance of a classification model. It provides a balanced assessment by calculating the harmonic mean of precision and recall. This single metric considers both recall and precision, offering a comprehensive evaluation of model performance. The F1-score is calculated as the harmonic mean of these two values, with the formula for the F-measure represented in equation 6.

$$F - measure = \frac{2 * \left(Precision \ * Recall\right)}{Precision \ + \ Recall} \tag{6}$$

The performance of a classifier can be effectively represented and evaluated using a confusion matrix. TP represents the number of individuals correctly identified as having the disease. True Negatives represent the number of individuals correctly identified as not having the disease. False Positives (FP) represent the number of healthy individuals who are incorrectly diagnosed with the disease. False Negatives (FN) occur when individuals with the disease are incorrectly classified as healthy.

4.2. Evaluative Results

The datasets are utilized to identify the most effective model for predicting cardiovascular disease in patients. This investigation is based on established algorithms commonly used in healthcare predictions. The nominated model demonstrated particularly strong performance in specific cases, making it a valuable tool for professionals in diagnosing this condition. All issues, comparisons, and outcomes are thoroughly illustrated in the subsequent tables.

Table 4 shows the performance evaluation of ML algorithms on the UCI dataset reveals notable differences in accuracy, precision, F1-score, log loss, and error rate. RF and DT achieved the highest accuracy (98.54%) and precision (0.9858), indicating superior performance. GB followed with 93.17% accuracy and balanced metrics, while SVM demonstrated strong results with an 88.78% accuracy. LR and GNB showed moderate performance, with accuracy values of 79.51% and 80%, respectively. KNN achieved 83.41% accuracy but slightly higher error rates compared to top-performing models. RF and DT stand out as optimal models for the dataset due to their low error rates (1.46%) and minimal log loss. These findings highlight RF and DT as robust classifiers for this dataset.

A comparison of error rates among seven ML algorithms, such as LR, DT, GNB, KNN, GB, RF, and SVM, is presented in Fig. 4. Among these, RF demonstrated the lowest error rate, showcasing its effectiveness and robustness for this dataset. GB and SVM also achieved strong results, with error rates slightly higher than that of RF. LR and GNB showed moderate performance, whereas KNN had a higher error rate. While the DT performed competitively, it was outperformed by the ensemble methods. Overall, ensemble

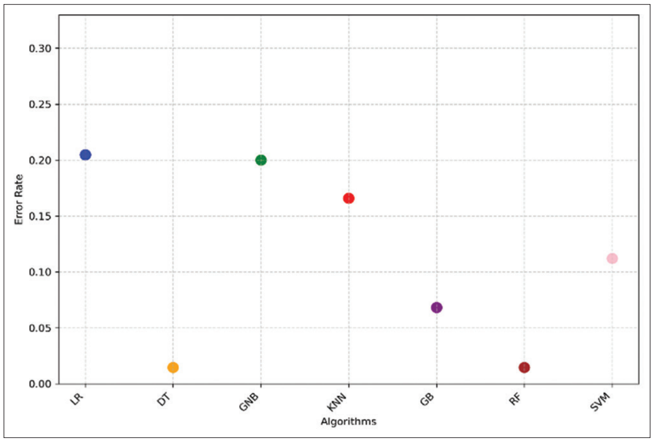


Fig. 4. Comparison of error rates for machine learning algorithms on UCI dataset.

TABLE 5: Performance metrics of classification algorithms on the UCI dataset using stratified sampling

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	80.97561	0.822476	0.807244	0.348282	0.190244
RF	100.00000	1.00000	1.00000	0.05885	0.00000
SVM	92.68293	0.927144	0.926787	0.178714	0.073171
GNB	82.92683	0.831469	0.828754	0.506077	0.170732
GB	97.56098	0.975649	0.975606	0.156275	0.02439
KNN	86.34146	0.863618	0.863434	0.216995	0.136585
DT	98.53659	0.985792	0.985368	0.527468	0.014634

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

models, especially RF, proved to be the most effective for this dataset.

To modify these algorithms as stated, a stratified procedure was employed. Table 5 presents the performance metrics of various classification algorithms applied to the UCI dataset using stratified sampling. The RF algorithm achieved perfect scores across all metrics, with 100% accuracy and zero error rates. DT closely followed with 98.54% accuracy and an error rate of 1.46%. GB performed impressively with 97.56% accuracy. SVM showed 92.68% accuracy, while KNN and GNB achieved accuracies of 86.34% and 82.93%, respectively. LR exhibited the lowest accuracy among the models at 80.98%. Overall, RF demonstrated superior performance, while LR lagged behind.

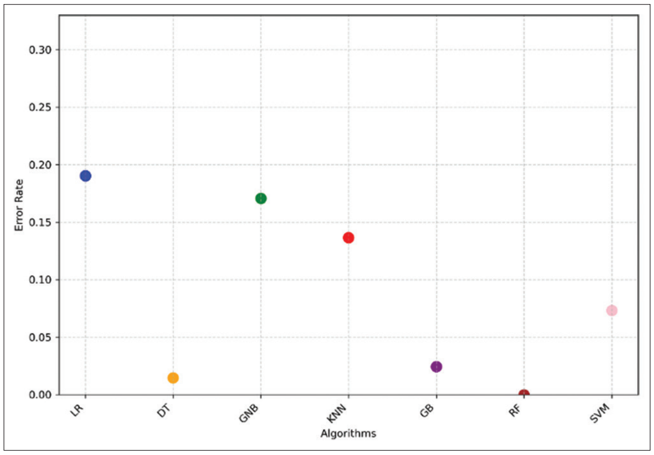


Fig. 5. Comparison of error rates for machine learning algorithms using stratified sampling on UCI dataset.

Moreover, Fig. 5 shows a comparison of error rates for various ML algorithms applied to the UCI dataset using stratified sampling. RF and DT achieve the lowest error rates, indicating the best performance among the evaluated methods. GB also exhibits a relatively low error rate, demonstrating strong predictive capabilities. SVM and KNN fall in the mid-range of error rates. LR and GNB show higher error rates, reflecting comparatively weaker performance. This comparison underscores the effectiveness of RF, DT, and GB in minimizing prediction errors.

Table 6 presents the performance metrics of various classification algorithms applied to the UCI dataset using k-fold cross-validation. LR achieved the highest accuracy of 84.16%, along with the best precision (0.8472) and F1-score (0.8392). RF followed with an accuracy of 81.52%, while SVM slightly outperformed RF in terms of precision (0.8292) and F1-score (0.8188), despite a marginally lower accuracy of 82.17%. GNB and GB exhibited similar performances, both with around 80.84% and 80.87% accuracy, respectively. KNN had a lower accuracy of 79.85% and the highest log loss (1.751), indicating weaker performance compared to other models. DT demonstrated the lowest accuracy (72.89%) and the highest log loss (9.773), making it the least effective model in this comparison. Overall, LR emerged as the most robust classifier, balancing accuracy, precision, and F1-score, while DT exhibited the weakest performance due to its high error rate and log loss.

Fig. 6 illustrates the error rates of various ML algorithms applied to the UCI dataset using k-fold cross-validation. The algorithms compared include LR, DT, GNB, KNN, GB, RF, and SVM. The error rate is plotted on the y-axis, while the different algorithms are labeled along the x-axis. For clarity, each algorithm is represented by a distinct color. The DT exhibits the highest error rate among the models, while LR achieves the lowest. The remaining models demonstrate relatively similar error rates, with slight variations. These results provide insights into the comparative performance of different classifiers on the given dataset, helping to determine the most effective model for classification tasks.

The performance metrics of classification algorithms on the HD dataset are shown in Table 7. KNN achieved the highest accuracy (90.16%) and the lowest error rate (9.83%), making it the top-performing algorithm in this study. SVM

TABLE 6: Performance metrics of classification algorithms on the UCI dataset using k-fold cross-validation

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	84.15847	0.847168	0.839233	0.408436	0.158415
RF	81.51913	0.819266	0.81326	0.404027	0.184809
SVM	82.17486	0.829186	0.818829	0.420519	0.178251
GNB	80.84153	0.8137	0.806308	0.580949	0.191585
GB	80.86885	0.811012	0.806967	0.476965	0.191311
KNN	79.84699	0.804062	0.794404	1.751016	0.20153
DT	72.88525	0.72869	0.728066	9.773148	0.271148

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

TABLE 7: Performance metrics of classification algorithms on the HD dataset

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	85.2459	0.853076	0.852538	0.364798	0.147541
RF	83.60656	0.836066	0.836066	0.3633	0.163934
SVM	86.88525	0.870862	0.868923	0.345582	0.131148
GNB	86.88525	0.870862	0.868923	0.693538	0.131148
GB	78.68852	0.787537	0.787	0.43882	0.213115
KNN	90.16393	0.903684	0.901692	1.96402	0.098361
DT	75.40984	0.770801	0.75224	8.863193	0.245902

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

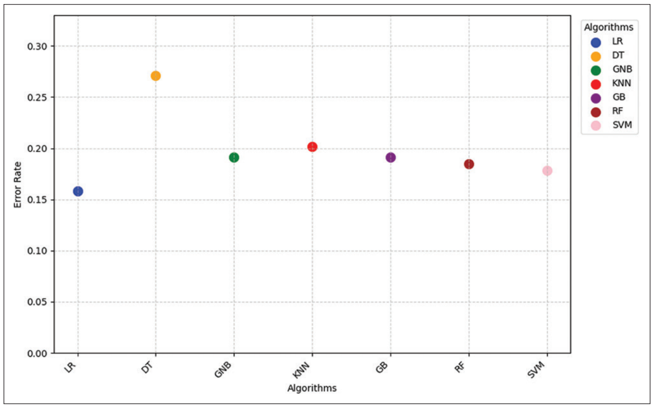


Fig. 6. Comparison of error rates across algorithms for UCI dataset using k-fold cross-validation.

and GNB both exhibited strong performance with identical accuracy (86.89%), precision (0.87), and F1-score (0.87). LR also performed well with an accuracy of 85.25%. RF showed moderate results with an accuracy of 83.61%, while DT achieved comparatively lower accuracy at 78.69% and 75.41%, respectively. The log loss values for SVM and LR were notably lower, indicating better calibration, while DT had the highest log loss, suggesting poorer reliability. These results position KNN as the most effective classifier for this dataset, with SVM and GNB following closely behind.

The classification results demonstrate significant performance variations across different algorithms. KNN achieved the lowest error rate, highlighting its effectiveness for this dataset.

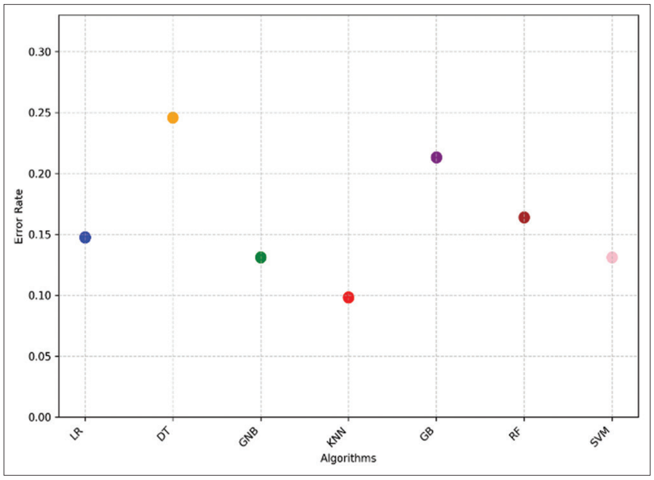


Fig. 7. Comparison of error rates across algorithms for HD dataset classification.

LR also performed well, with relatively low error rates. In contrast, DT exhibited the highest error rate, reflecting its lower suitability for this task. RF and GB delivered intermediate performance, balancing error rates and model complexity. SVM showed moderate performance, while GNB lagged behind the top-performing algorithms but outperformed DT. These findings, as depicted in Fig. 7, underscore the importance of selecting an appropriate algorithm to enhance classification accuracy.

The performance metrics of classification algorithms on the HD dataset using stratified sampling are presented in Table 8,

TABLE 8: Performance metrics of classification algorithms on the HD dataset using stratified sampling

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	80.32787	0.812564	0.799672	0.438066	0.196721
RF	83.60656	0.858297	0.831266	0.404595	0.163934
SVM	81.96721	0.834563	0.815437	0.425831	0.180328
GNB	81.96721	0.826237	0.817182	0.61126	0.180328
GB	81.96721	0.826237	0.817182	0.449421	0.180328
KNN	80.32787	0.812564	0.799672	2.069169	0.196721
DT	70.4918	0.705287	0.702	10.63583	0.295082

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

highlighting key insights. LR achieved an accuracy of 80.33%, a precision of 0.813, and an F1-score of 0.800, with a log loss of 0.438 and an error rate of 0.197. RF outperformed others with the highest accuracy of 83.61%, precision of 0.858, and an F1-score of 0.831, while maintaining a log loss of 0.405 and an error rate of 0.164. SVM and GB both reached an accuracy of 81.97%, with comparable precision and F1-Scores, though GB had slightly better log loss at 0.449. KNN and GNB had identical accuracies of 80.33% and 81.97%, respectively, but KNN showed significantly higher log loss at 2.069. DT performed the worst, with an accuracy of 70.49% and a log loss of 10.636, reflecting its limitations compared to other models.

Fig. 8 compares the error rates of various classification algorithms for the HD dataset using stratified sampling. DT exhibited the highest error rate, approximately 0.30, highlighting its relatively poor performance. LR, KNN, and GNB demonstrated similar error rates around 0.20, indicating moderate performance. GB, RF, and SVM achieved lower error rates, with RF standing out as the most accurate model, achieving an error rate of approximately 0.16. This emphasizes the effectiveness of ensemble methods, such as RF and GB in reducing classification errors compared to simpler models such as DT.

Table 9 presents the performance metrics of various classification algorithms evaluated on the HD dataset using k-fold cross-validation. The DT algorithm achieved the highest performance, attaining 100% accuracy, precision, and F1-score, with a log loss of 0 and an error rate of 0. RF closely followed, exhibiting an accuracy of 99.61%, precision of 0.9962, and an F1-score of 0.9961, with minimal log loss (0.0516) and a very low error rate (0.0039). GB also demonstrated strong performance, achieving 97.17% accuracy, a precision of 0.9722, and an F1-score of 0.9717, with a log loss of 0.1364 and an error rate of 0.0283. The SVM classifier achieved an accuracy of 92.39% and a precision of 0.9241, with an F1-score of 0.9239 and a moderate log loss of 0.1991. LR and KNN exhibited comparable performance, with 84.59% and 83.61% accuracy, respectively. LR showed a slightly higher precision (0.8506) and F1-score (0.8451) compared to KNN (0.8380 precision, 0.8359 F1-score). GNB recorded the lowest accuracy (82.63%) and the highest log loss (0.5122), indicating weaker reliability than other models. Overall, DT and RF emerged as the most effective classifiers, with GB also demonstrating competitive performance.

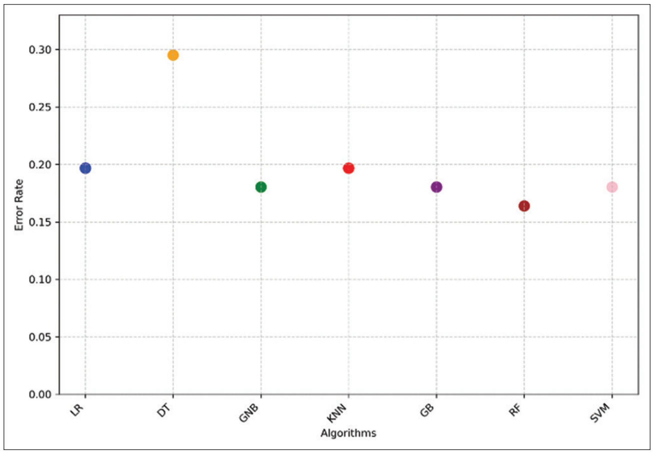


Fig. 8. Comparison of error rates across algorithms for HD dataset classification using stratified sampling.

Fig. 9 presents a comparative analysis of the error rates for various ML algorithms applied to the HD dataset using k-fold cross-validation. The x-axis represents different algorithms, including LR, DT, GNB, KNN, GB, RF, and SVM. The y-axis denotes the corresponding error rates. For clarity, each algorithm is represented by a distinct color. The results indicate that DT exhibits the lowest error rate, demonstrating its strong classification performance for this dataset. In addition, RF and GB also show relatively low error rates, reinforcing the effectiveness of ensemble methods. Conversely, GNB and KNN yield higher error rates, while LR and SVM fall in between. These findings highlight the

TABLE 9: Performance metrics of classification algorithms on the HD dataset using k-fold cross-validation

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	84.58537	0.850558	0.845109	0.362773	0.154146
RF	99.60976	0.996248	0.996098	0.051581	0.003902
SVM	92.39024	0.924054	0.923882	0.199138	0.076098
GNB	82.63415	0.828886	0.825682	0.512226	0.173659
GB	97.17073	0.97216	0.971708	0.136384	0.028293
KNN	83.60976	0.837965	0.835948	0.222265	0.163902
DT	100	1	1	0	0

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

TABLE 10: Performance metrics of classification algorithms on the UCI-HD dataset

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	83.08271	0.838173	0.829144	0.377472	0.169173
RF	100.00000	1.000000	1.000000	0.025382	0.000000
SVM	92.85714	0.930338	0.928379	0.183863	0.071429
GNB	82.70677	0.832169	0.825731	0.543392	0.172932
GB	96.99248	0.970311	0.969898	0.128287	0.030075
KNN	94.73684	0.94913	0.947392	0.129948	0.052632
DT	100.0000	1.0000	1.0000	2.22E-16	0.000000

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

performance variations among different models and suggest that DT is particularly well-suited for this dataset.

Significantly, this model was modified to show that the combination of these two datasets affected its performance. Therefore, Table 10 presents the performance metrics of various classification algorithms on the UCI-HD dataset, offering insights into their effectiveness. DT and RF both achieved perfect performance with 100% accuracy, precision, and F1-score, along with zero error rates and minimal log loss. GB exhibited excellent results, with 96.99% accuracy and a precision of 0.970, followed by KNN at 94.73% accuracy. SVM showed strong performance, achieving 92.86% accuracy and a log loss of 0.1838. LR and GNB demonstrated moderate effectiveness, with accuracies of 83.08% and 82.71%, respectively. The table highlights the dominance of ensemble methods and DT in classification tasks.

The error rates of various ML algorithms applied to the UCI Heart Disease (UCI-HD) dataset are illustrated in Fig. 10. RF and DT achieved the lowest error rates, showcasing their strong predictive performance. GB also performed competitively, with slightly higher error rates. KNN algorithm demonstrated moderate accuracy, while LR and GNB produced comparatively higher error rates, indicating reduced effectiveness.

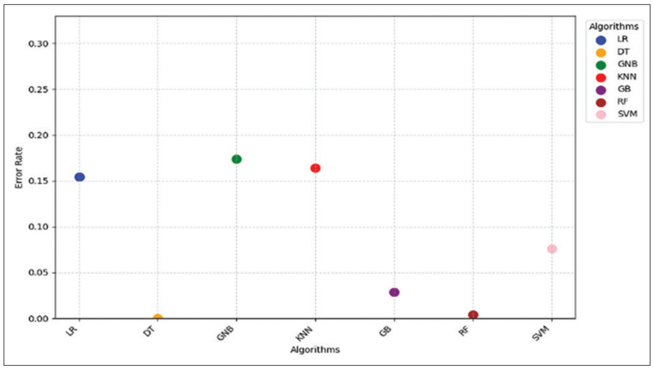


Fig. 9. Comparison of error rates across algorithms for the HD dataset using k-fold cross-validation.

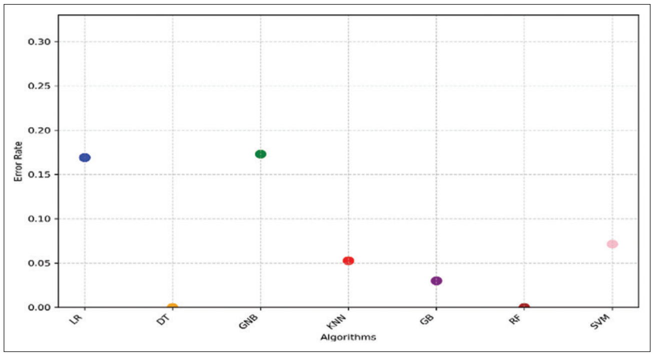


Fig. 10. Comparison of error rates for machine learning algorithms on the UCI-HD dataset.

The results clearly determine the performance metrics of various classification algorithms applied to the UCI-HD dataset using stratified sampling, as shown in Table 11. DT and RF achieved perfect results, with 100% accuracy,

TABLE 11: Performance metrics of classification algorithms on UCI-HD dataset using stratified sampling

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	84.58647	0.847779	0.845358	0.352075	0.154135
RF	100.00000	1.000000	1.000000	0.021479	0.000000
SVM	95.11278	0.95114	0.95112	0.140181	0.048872
GNB	84.21053	0.842237	0.84198	0.47266	0.157895
GB	98.1203	0.981225	0.9812	0.12264	0.018797
KNN	96.61654	0.96678	0.966129	0.116318	0.033835
DT	100.00000	1.000000	1.000000	2.22E-16	0.000000

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

precision, and F1-score, and zero error rates. GB and KNN also demonstrated high performance, with accuracies of 98.12% and 96.62%, respectively. SVM showed strong results, achieving 95.11% accuracy. LR and GNB exhibited moderate performance, with accuracies of 84.59% and 84.21%. The metrics highlight the superior predictive capabilities of ensemble models, such as RF and DT.

The results indicate that for the stratified modification, the error rates of various classification algorithms applied to the UCI-HD dataset using stratified sampling are illustrated in Fig. 11. The comparison reveals that SVM achieved the lowest error rate, indicating its superior performance in accurately classifying the dataset. RF and GB also exhibited competitive error rates, showcasing their effectiveness in handling the dataset's complexity. Conversely, algorithms such as GNB demonstrated relatively higher error rates, suggesting challenges in capturing the underlying data patterns. The stratified sampling methodology ensured balanced class representation, which contributed to the robustness of the evaluation. These results emphasize the need for selecting appropriate algorithms for high-stakes applications, such as heart disease prediction, where classification accuracy is paramount.

Table 12 presents the performance metrics of various classification algorithms evaluated on the UCI-HD dataset using k-fold cross-validation. The DT and RF classifiers achieved perfect accuracy (100%) with an error rate of 0, demonstrating their strong predictive capabilities. The GB classifier followed closely with an accuracy of 98.12%, a precision of 0.981, and an F1-score of 0.981, indicating robust performance. The SVM model also exhibited high accuracy (94.88%) and precision (0.949), while KNN performed slightly lower with an accuracy of 93.53%. LR and GNB recorded relatively lower accuracy scores of 85.16% and 82.91%, respectively, with GNB having the highest log loss value (0.5147), indicating more significant uncertainty in its predictions. DT achieved the lowest log loss among all models, emphasizing its reliability. Overall, ensemble-based methods (RF and GB) outperformed other classifiers in terms of accuracy and precision, highlighting their effectiveness in heart disease classification.

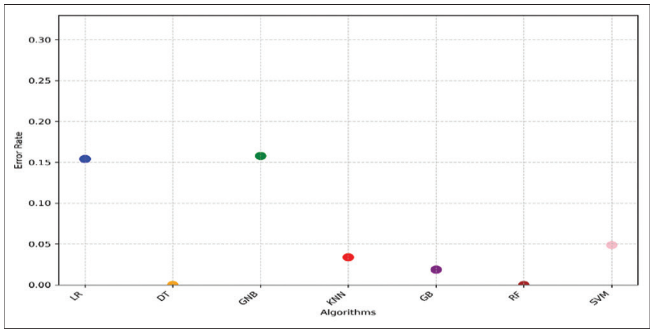


Fig. 11. Comparison of error rates across algorithms for UCI-HD dataset classification using stratified sampling.

Fig. 12 presents a comparative analysis of the error rates for different ML algorithms on the UCI-HD dataset using k-fold cross-validation. The x-axis represents the algorithms evaluated, including LR, DT, GNB, KNN, GB, RF, and SVM. The y-axis denotes the corresponding error rates. Each algorithm is color-coded for clarity, as shown in the legend. The results indicate that RF and GB achieved the lowest error rates, suggesting superior predictive performance, while GNB and LR exhibited higher error rates. The variability in error rates highlights the importance of model selection in achieving optimal classification performance on this dataset.

5. DISCUSSION

There are differences in the evaluation models and results obtained from the combined datasets or stratified algorithms compared to previous studies on the same datasets. All relevant studies are referenced in the background review. However, the variations in performance highlight the superior results achieved by specific algorithms, as noted.

TABLE 12: Performance metrics of classification algorithms on the UCI-HD dataset using k-fold cross-validation

Algorithm	Accuracy	Precision	F1-score	Log loss	Error rate
LR	85.16385303	0.857201168	0.850471332	0.361969257	0.14836147
RF	100	1	1	0.020635519	0
SVM	94.87955738	0.949310038	0.948753901	0.15644934	0.051204426
GNB	82.90707902	0.830871801	0.828611865	0.514689846	0.17092921
GB	98.11717974	0.981378517	0.981166354	0.123583547	0.018828203
KNN	93.52617393	0.936109038	0.935228791	0.1410065	0.064738261
DT	100	1	1	0	0

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

Table 13 presents the accuracy of several classification algorithms across the UCI, HD, and combined UCI-HD datasets. LR achieved accuracies of 79.512%, 85.246%, and 83.083% for the UCI, HD, and UCI-HD datasets, respectively. RF demonstrated the highest accuracy on the UCI dataset (98.537%) but showed a decline on the HD dataset (83.607%), achieving a perfect accuracy of 100% on the combined dataset. SVM recorded 88.78% for UCI, 83.607% for HD, and 92.857% for UCI-HD. GNB achieved 80% on UCI, 86.885% on HD, and 82.707% on the combined dataset. GB performed with 93.171% on UCI, 78.689% on HD, and 96.992% on UCI-HD. KNN achieved 83.415% on UCI, 90.164% on HD, and 94.737% on the combined dataset. Finally, DT reached an accuracy of 98.537% on UCI, 75.41% on HD, and 100% on UCI-HD. Overall, RF and DT demonstrated the best performance on the combined UCI-HD dataset, achieving perfect accuracy. However, other algorithms, such as GB and KNN, also showed notable effectiveness across the datasets.

The comparison in Table 14 highlights the impact of using stratification on algorithm accuracy with the UCI dataset. Without stratification, the accuracy varied across algorithms, with DT and RF achieving the highest accuracies at 98.537%, reflecting their strong classification capabilities. GB also performed well at 93.171%, while SVM, KNN, and GNB exhibited moderate accuracies of 88.780%, 83.415%, and 80%, respectively. LR showed the lowest accuracy at 79.512%, indicating potential limitations with non-stratified data. Stratification improved accuracy for most algorithms, ensuring better representation of class distributions during training. RF achieved a perfect 100% accuracy with stratification, while GB, SVM, and KNN also demonstrated significant gains. DT performance remained constant, indicating minimal dependency on class distribution in this case.

Table 15 compares algorithm accuracies on the HD dataset with and without stratified sampling. Most algorithms

TABLE 13: Accuracy of classification algorithms on UCI, HD, and combined UCI-HD datasets

Algorithm	Accuracy UCI	Accuracy HD	Accuracy UCI-HD
LR	79.512	85.246	83.083
RF	98.537	83.607	100
SVM	88.78	83.607	92.857
GNB	80	86.885	82.707
GB	93.171	78.689	96.992
KNN	83.415	90.164	94.737
DT	98.537	75.41	100

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

TABLE 14: Comparison of algorithm accuracies with and without stratification according to the UCI dataset

Algorithm	Accuracy	Accuracy (Stratify=y)
LR	79.512	80.976
RF	98.537	100.000
SVM	88.780	92.683
GNB	80.000	82.927
GB	93.171	97.560
KNN	83.415	86.341
DT	98.537	98.537

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

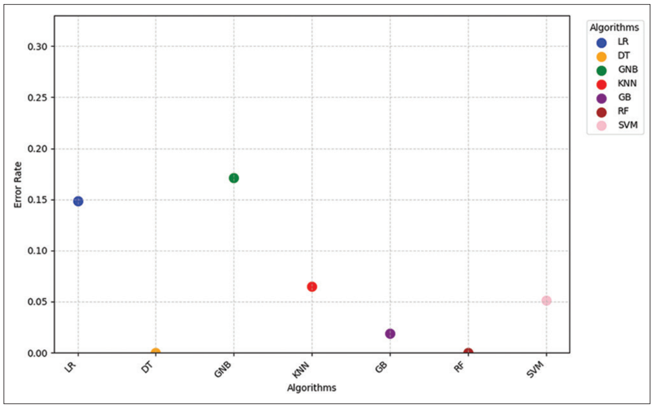


Fig. 12. Comparison of error rates across algorithms for the UCI-HD dataset using k-fold cross-validation.

showed a decline in accuracy with stratified sampling, such as KNN (90.164–80.328%) and LR (85.246–80.328%), indicating sensitivity to data redistribution. RF maintained consistent accuracy (83.607%) across both cases, showcasing its robustness. GB slightly improved (78.689–81.967%), suggesting enhanced generalization. Other algorithms, such as SVM and GNB, experienced moderate declines, indicating varied sensitivity. These results highlight that stratified sampling impacts algorithms differently, emphasizing the need for careful evaluation of sampling strategies for optimal performance.

The results in Table 16 highlight the impact of stratified sampling on algorithm accuracy when applied to the UCI-HD dataset. The majority of algorithms show an improvement in accuracy with stratified sampling, notably LR, which increases from 83.083% to 84.586%, and GNB, which improves from 82.707% to 84.211%. Similarly, KNN shows a notable increase from 94.737% to 96.617%, and GB improves from 96.992% to 98.120%. SVM also sees an increase from 92.857% to 95.113%. In contrast, DT and RF algorithms maintain their perfect accuracy of 100% with and without stratification. These findings suggest that stratified sampling can provide slight accuracy enhancements, particularly for algorithms that initially perform below perfect accuracy, while having no effect on algorithms already achieving optimal results. This suggests the potential of stratified sampling to improve model performance, especially for classifiers sensitive to class imbalances.

TABLE 15: Algorithm accuracies: Comparison with and without stratified sampling HD dataset

Algorithm	Accuracy	Accuracy (Stratify=y)
LR	85.246	80.328
RF	83.607	83.607
SVM	83.607	83.607
GNB	86.885	81.967
GB	78.689	81.967
KNN	90.164	80.328
DT	75.410	70.492

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

TABLE 16: Algorithm accuracies: Comparison with and without stratified sampling UCI – HD dataset

Algorithm	Accuracy	Accuracy (Stratify=y)
LR	83.083	84.586
RF	100	100
SVM	92.857	95.113
GNB	82.707	84.211
GB	96.992	98.120
KNN	94.737	96.617
DT	100	100

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

In previous experiments with the UCI and HD datasets, several machines learning algorithms, including LR, RF, SVM, GNB, and KNN, were evaluated. For the UCI dataset, RF achieved the highest accuracy of 98.53%, followed by SVM at 87.31%. LR reached an accuracy of 82.92%, while KNN recorded 81.95%. GNB showed the lowest accuracy at 74.63% [34]. Notably, our modifications resulted in improved accuracy across these models compared to previous results.

Table 17 presents the classification accuracy of seven algorithms, LR, RF, SVM, GNB, GB, KNN, and DT, evaluated on the UCI, HD, and combined UCI-HD datasets using stratified sampling. Among these, RF and DT achieved perfect accuracy (100%) on both the UCI and UCI-HD datasets, indicating excellent performance. SVM also delivered strong results, with accuracy scores of 92.68% on UCI and 95.11% on UCI-HD, closely followed by GB with 97.56% and 98.12%, respectively.

However, DT showed a sharp decline in performance on the HD dataset, recording the lowest accuracy of 70.49%, while RF maintained a comparatively higher score of 83.61%. This suggests that the HD dataset introduced more complex classification challenges for tree-based models. Overall, the combined UCI-HD dataset resulted in improved accuracy for most algorithms, demonstrating the advantage of data integration. LR, GNB, and KNN exhibited moderate yet stable performance across all datasets, with accuracies ranging from 80.33% to 96.62%. These findings underscore the superior generalization ability of ensemble methods such as RF and GB, particularly when applied to enriched and well-balanced datasets.

The exceptionally high accuracy (100%) observed for the RF and DT models, particularly on the combined UCI-HD dataset with stratified sampling, may initially raise concerns of overfitting. However, several factors inherent to the study provide a reasonable justification for this performance. First, the application of stratified sampling ensured balanced class representation across both training and testing sets, thereby minimizing the risk of class imbalance-related bias. Second, the datasets underwent rigorous pre-processing, which eliminated missing values and ensured well-defined feature distributions, likely enhancing the learning capacity of tree-based algorithms. Third, the combined dataset incorporated consistent features from two closely related sources

TABLE 17: Accuracy of classification algorithms on UCI, HD, and combined UCI-HD datasets using stratified sampling

Algorithm	Accuracy UCI (Stratify=y)	Accuracy HD (Stratify=y)	Accuracy UCI-HD (Stratify=y)
LR	80.976	80.328	84.586
RF	100.000	83.607	100
SVM	92.683	83.607	95.113
GNB	82.927	81.967	84.211
GB	97.560	81.967	98.120
KNN	86.341	80.328	96.617
DT	98.537	70.492	100

LR: Logistic regression, RF: Random forest, SVM: Support vector machine, GNB: Gaussian Naive Bayes, GB: Gradient boosting, KNN: K-nearest neighbors, DT: Decision tree

(UCI and HD), which may have facilitated more distinct classification boundaries. Moreover, the RF, as an ensemble method, mitigates overfitting by averaging the outputs of multiple decorrelated DTs. While perfect accuracy warrants cautious interpretation, the results remained consistent across various datasets and sampling techniques, suggesting that the model's performance is robust rather than indicative of data memorization. Nonetheless, to further confirm generalizability, additional validation on external datasets or through k-fold cross-validation would be a valuable next step in future work. It is also possible that data augmentation was employed to prevent overfitting.

The study evaluated ML models for CVD prediction, showing distinct performance patterns. Ensemble models, such as RF, GB, and DT were top performers, with RF achieving perfect accuracy, especially using stratified sampling. Stratification was crucial for improving SVM and GNB by maintaining class balance and reducing bias. LR showed moderate performance due to its linear limitations, while SVM excelled in high-dimensional spaces. KNN performed well on smaller datasets but struggled with larger, complex ones. GNB was competitive but limited by its Gaussian assumptions. Combining UCI and HD datasets enhanced performance for all models, especially ensemble methods. RF and DT achieved 100% accuracy on the combined dataset. Despite perfect accuracy, safeguards, such as pre-processing and stratification minimized overfitting risks. The findings highlight the value of ensemble methods, dataset integration, and stratified sampling, suggesting future work on hybrid models and real-world validation.

Fig. 13 displays the classification accuracy of seven ML algorithms, LR, RF, SVM, GNB, GB, KNN, and DT, evaluated using k-fold cross-validation. Performance is compared across three datasets: UCI, HD, and their merged variant (UCI-HD). Notably, RF and DT achieved perfect accuracy (100%) on the HD and UCI-HD datasets, while LR and SVM demonstrated consistently high performance across all datasets. The figure highlights the impact of dataset variation on model accuracy, underscoring the robustness of ensemble methods, such as RF and GB.

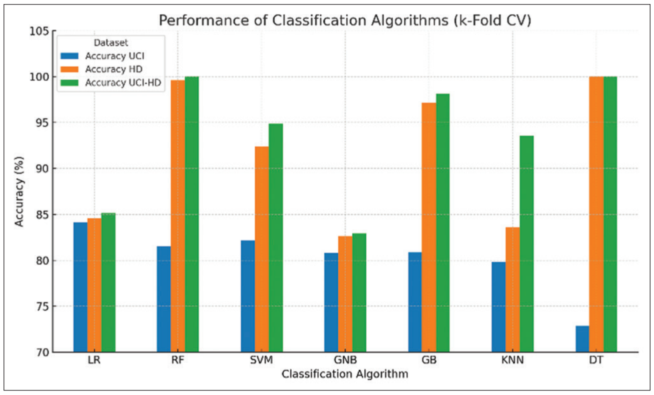


Fig. 13. Comparison of classification algorithm performance across UCI, HD, and UCI-HD datasets using k-Fold cross-validation.

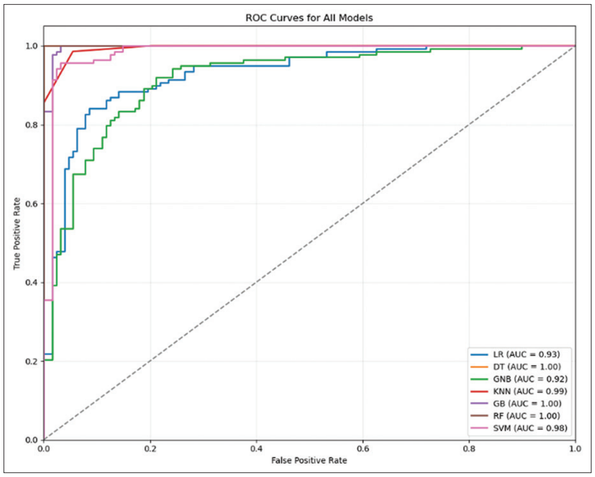


Fig. 14. ROC curve evaluation of machine learning models: Ensemble methods achieve perfect AUC.

Fig. 14 illustrates the Receiver Operating Characteristic (ROC) curves for all evaluated classification models, providing a

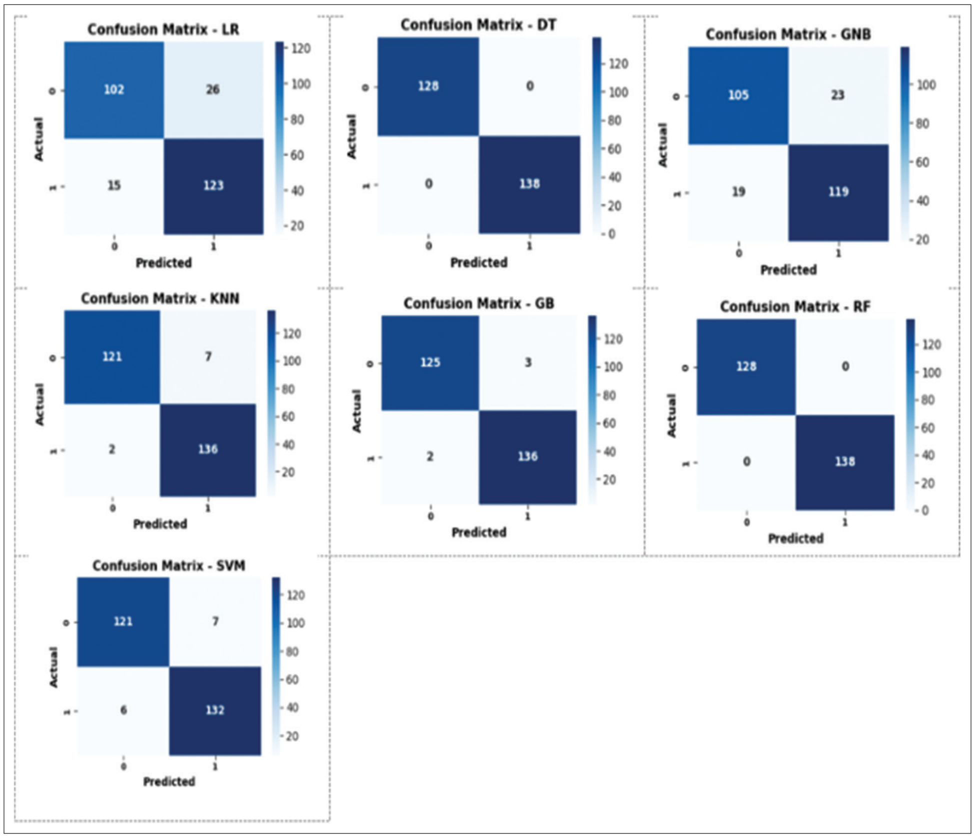


Fig. 15. Confusion matrix comparison of ML models.

visual comparison of their diagnostic performance. The Area Under the Curve (AUC) metric is presented for each model, with higher values indicating better discriminative ability. Ensemble methods, such as DT, GB, and RF achieved perfect classification performance with an AUC of 1.00. The KNN and SVM models also demonstrated excellent performance with AUCs of 0.99 and 0.98, respectively. LR and GNB had slightly lower AUCs of 0.93 and 0.92, but still showed strong classification capability. Overall, the ROC analysis confirms the superior performance of ensemble methods and supports their robustness in distinguishing between classes with minimal false positive rates.

The comparative analysis of confusion matrices, as illustrated in Fig. 15, reveals substantial differences in classification performance across the evaluated models. Both DT and RF classifiers achieved perfect predictive accuracy (100%) with F1-scores of 1.00, indicating zero misclassifications for both negative (class 0) and positive (class 1) instances, 128 and 138 samples, respectively. GB also demonstrated exceptional performance, attaining an accuracy of 98.9% and an F1-score of 0.989, with only five misclassifications (three FP and two FN). Similarly, the KNN classifier performed strongly, with 97.8% accuracy and a 0.978 F1-score, misclassifying seven negative and two positive cases.

The SVM model yielded commendable results with 96.9% accuracy and a 0.969 F1-score, though it incurred slightly higher misclassification rates (seven FP and six FN). In contrast, LR and GNB underperformed relative to the other models. LR achieved 89.9% accuracy and a 0.896 F1-score, misclassifying 26 negative and 15 positive instances. GNB recorded the lowest performance, with an accuracy of 87.0% and an F1-score of 0.873, resulting in 42 total misclassifications.

These results, as depicted in Fig. 15, underscore the superiority of ensemble methods, particularly RF and GB, in effectively capturing complex patterns within the dataset. Conversely, simpler linear (LR) and probabilistic GNB models may be less capable in such high-dimensional classification tasks.

6. CONCLUSION

CVD remains a leading cause of mortality worldwide, emphasizing the critical need for early diagnosis and intervention. If the symptoms of heart disease are not promptly identified and treated, the condition can escalate into life-threatening scenarios. Artificial intelligence has been effectively utilized for CVD prediction, with advancements promising increasingly accurate forecasts based on historical medical data. Despite significant progress in this domain, continuous enhancements in predictive methodologies are essential and highly encouraged.

In conclusion, these modifications improved CVD prediction by utilizing seven ML algorithms: LR, RF, SVM, GNB, GB, KNN, and DT. By analyzing medical histories of patients with severe heart conditions, the models classified individuals based on their risk of developing CVD. The models were trained and tested on datasets containing factors such as chest discomfort, high blood pressure, and cardiac arrest. To ensure robustness, evaluations were performed with and without the stratify parameter. The results revealed that the DT and RF algorithms consistently achieved the peak accuracy rates, with both models reaching 100% accuracy on the combined dataset. Besides, the stratified technique enhanced the accuracy across all methods. These findings emphasize the critical role of sufficient training data and stratification in improving predictive performance. They also highlight the potential of AI-driven tools to assist healthcare professionals in making faster and more accurate diagnoses, ultimately lowering costs and enhancing patient outcomes. The results represent a significant advancement in the field by achieving higher accuracy rates than previous studies, setting a standard for the practical application of ML in CVD prediction. Among the tested methods, DT and RF emerged as the most reliable, highlighting the efficiency of ensemble learning techniques in clinical applications.

This research inspires future work to explore integrating additional datasets, refining algorithms, and developing real-time prediction systems to further advance the field. In addition, the combined dataset (UCI-HD) was tested using novel classifier metaheuristic algorithms, such as the fitness dependent optimizer (FDO) [35], [36].

REFERENCES

[1] X. Han. "Heart Disease Type Prediction Model Based on SVM-ANN". In: *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, pp. 422-426, 2022.

[2] A. R. Snigdha, S. N. Tasnim, K. R. Miah and T. Islam. "Early Prediction of Heart Attack using Machine Learning Algorithms". In: *Proceedings of the 2nd International Conference on Computing Advancements*, pp. 344-348, 2022.

[3] A. Lahsasna, R. N. Ainon, R. Zainuddin and A. Bulgiba. "Design of a fuzzy-based decision support system for coronary heart disease diagnosis". *Journal of Medical Systems*, vol. 36, pp. 3293-3306, 2012.

[4] S. Song, T. Chen and G. Antoniou. "ANFIS Models for Heart Disease Prediction". In: *Proceedings of the 2021 5th International Conference on Innovation in Artificial Intelligence*", pp. 32-35, 2021.

[5] T. Suresh, T. A. Assegie, S. Rajkumar and N. Komal Kumar. "A hybrid approach to medical decision-making: Diagnosis of heart disease with machine-learning model". *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, p. 1831, 2022.

[6] A. A. Hussein. "Improve the performance of K-means by using genetic algorithm for classification heart attack". *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 2, p. 1256, 2018.

[7] K. Wang, J. Tian, C. Zheng, H. Yang, J. Ren, Y. Liu and Q. Han, Y. Zhang. "Interpretable prediction of 3-year all-cause mortality in patients with heart failure caused by coronary heart disease based on machine learning and SHAP". *Computers in Biology and Medicine*, vol. 137, p. 104813, 2021.

[8] S. Geetha, C. P. Devi, V. Kalaivani, C. J. Haritha and G. Preetha. "Prediction techniques of heart disease and diabetes disease using machine learning". *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 10, pp. 3316-3325, 2021.

[9] D. O. Hasan and A. M. Aladdin. "Sleep-related consequences of the COVID-19 pandemic: A survey study on insomnia and sleep apnea among affected individuals". *Insights in Public Health Journal*, vol. 5, no 2, 2024.

[10] R. K. Muhammed, R. R. Aziz, A. A. Hassan, A. M. Aladdin, S. J. Saydahet and T. A. Rashidal. "Comparative analysis of AES, blowfish, twofish, salsa 20, and ChaCha20 for image encryption". *Kurdistan Journal of Applied Research*, vol. 9, no. 1, pp. 52-65, 2024.

[11] Z. Rayan, M. Alfonse and A. B. M. Salem. "Machine learning approaches in smart health". *Procedia Computer Science*, vol. 154, pp. 361-368, 2019.

[12] A. M. Aladdin and T. A. Rashid. "*Leo: Lagrange Elementary Optimization*". Germany, Springer, 2024.

[13] A. M. Aladdin and T. A. Rashid. "A new lagrangian problem crossover-a systematic review and meta-analysis of crossover standards". *Systems*, vol. 11, no. 3, p. 144, 2023.

[14] R. Mohammed, N. K. Al-Salihi, T. A. Rashid, A. M. Aladdin, M. Mohammadi and J. Majidpour. "*Artificial Cardiac Conduction System: Simulating Heart Function for Advanced Computational Problem Solving*". [Preprint], 2024.

[15] A. Budianto, R. Ariyuana, and D. Maryono, "Perbandingan K-Nearest Neighbor (Knn) Dan Support Vector Machine (Svm) Dalam Pengenalan Karakter Plat Kendaraan Bermotor," Jurnal Universitas Sebelas Maret, vol. 11, no. 1, p. 27, Nov. 2019, doi: 10.20961/jiptek.v11i1.18018.

[16] A. Gavhane, G. Kokkula, I. Pandya and K. Devadkar. "Prediction of Heart Disease using Machine Learning". In: *2018nd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, 2018, pp. 1275-1278.

[17] S. Ambekar and R. Phalnikar. "Disease Risk Prediction by Using Convolutional Neural Network". In: *2018 4th International Conference on Computing Communication Control and Automation (ICCUBEA)*, IEEE, 2018, pp. 1-5.

[18] N. Jothi, W. Husain, N. A. Rashid and S. Syed-Mohamad.

“Feature selection method using genetic algorithm for medical dataset”. *International Journal on Advanced Science Engineering Information Technology*, vol. 9, no. 6, pp. 1907-1912, 2019.

[19] T. A. Assegie. “A support vector machine based heart disease prediction”. *Journal of Software Engineering and Intelligent Systems*, vol. 4, pp. 111-116, 2019.

[20] E. S. Kajal and M. Nishika. “Prediction of heart disease using data mining techniques”. *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 2, no. 3, pp. 1-7, 2016.

[21] S. Babu, E. M. Vivek, K. P. Famina, K. Fida, P. Aswathi, M. Shanid and M. Hena. “Heart Disease Diagnosis using Data Mining Technique”. In: 2017 *International Conference of Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2017, pp. 750-753.

[22] R. Kannan and V. Vasanthi. “Machine Learning Algorithms with ROC Curve for Predicting and Diagnosing the Heart Disease”. In: N. B. Muppalaneni, M. Ma and S. Gurumoorthy, Eds. *Soft Computing and Medical Bioinformatics*, Springer, Singapore, 2019, pp. 63-72.

[23] K. Raza. “Improving the Prediction Accuracy of Heart Disease with Ensemble Learning and Majority Voting Rule”. In: *U-Healthcare Monitoring Systems*. Academic Press, United States, 2019, pp. 179-196.

[24] L. Sapra, J. K. Sandhu and N. Goyal. “Intelligent method for detection of coronary artery disease with ensemble approach”. In: *Advances in Communication and Computational Technology: Select Proceedings of ICACCT 2019*. Springer, 2021, pp. 1033-1042.

[25] A. Al Ahdal, M. Rakhra, R. R. Rajendran, F. Arslan, M. A. Khder, B. Patel and B. R. Rajagopal, R. Jain. “Monitoring cardiovascular problems in heart patients using machine learning”. *Journal of Healthcare Engineering*, vol. 2023, no. 1, p. 9738123, 2023.

[26] S. Patidar, A. Jain and A. Gupta. “Comparative Analysis of Machine Learning Algorithms for Heart Disease Predictions”. In: *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2022, pp. 1340-1344. doi: 10.1109/ICICCS53718.2022.9788408

[27] N. S. Noori, B. H. Hameed, and M. Kh. Mohammed, “An economic evaluation of the performance efficiency of conservation agriculture and food security projects using logistic regression in iraq for the 2022-2023 season,” anbar journal of agricultural sciences, vol. 22, no. 2, pp. 1033–1049, Dec. 2024, doi: 10.32649/ajas.2024.184466.

[28] Y. Chen, L. Li, W. Li, Q. Guo, Z. Du and Z. Xu. “Fundamentals of neural networks”. *AI Computing Systems*. Elsevier, Netherlands, pp. 17-51, 2024.

[29] M. Schonlau and R. Y. Zou. “The random forest algorithm for statistical learning”. *The Stata Journal: Promoting Communications on Statistics and Stata*, vol. 20, no. 1, pp. 3-29, 2020.

[30] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir and R. Sun. “A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms”. *Mobile Information Systems*, vol. 2018, no. 1, p. 3860146, 2018.

[31] S. Naiem, A. E. Khedr, A. M. Idrees and M. I. Marie. “Enhancing the efficiency of gaussian naïve bayes machine learning classifier in the detection of DDOS in cloud computing”. *IEEE Access*, vol. 11, pp. 124597-124608, 2023.

[32] M. Malohlava and A. Candel. “*Gradient Boosting Machine with H2O*”. H20 Booklet, 2016. Available from: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/booklets [Last accessed on 2025 Apr 04].

[33] I. Maryani, Rousyati, Indriyanti, D. Pratmanto, Y. M. Kristania and M. Maulidah. “Prediction of Heart Disease using Decision Tree in Comparison with Particle Swarm Optimization to Improve Accuracy”. In: *Proceedings of the 3rd International Conference on Advanced Information Scientific Development, SCITEPRESS - Science and Technology Publications*, 2023, pp. 233-239.

[34] S. Patidar, D. Kumar and D. Rukwal. Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction”. In: *ITM Web of Conferences*, 2022. doi: 10.3233/ATDE220723

[35] A. M. Aladdin and A. M. Abdulla. “Fitness-Dependent Optimizer for IoT Healthcare Using Adapted Parameters: A Case Study Implementation”. In: *Practical Artificial Intelligence for Internet of Medical Things*, CRC Press, United States, 2023, pp. 45-61.

[36] J. M. Abdullah and T. Ahmed. “*Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process*”. Vol. 7. IEEE Access, Park Avenue, pp. 43473-43486, 2019.