RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

HEART DISEASE PREDICTION SYSTEM

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (IS353IA)

V SEMESTER

OPEN-ENDED PROJECT REPORT

Submitted by

MANASA D N	**1RV22CS104**
MANVITHA H	**1RV23CS408**

Under the guidance of

Dr. Vinay V Hegde
Professor,
Dept of CSE, RVCE.

Dr. Veena Gadad
Professor,
Dept of CSE, RVCE.

Bachelor of Engineering in

Computer Science and Engineering

2024-2025

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the **Artificial Intelligence and Machine Learning Open-Ended Project Work** titled **"Heart Disease(Cardiac Arrest)Risk Prediction System"** is carried out by **Manasa D N (1RV22CS104) and Manvitha H (1RV23CS408)** who are bonafide student/s of RV College of Engineering, Bengaluru, in partial fulfillment for the **Internal Assessment of Course: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (IS353IA)** during the year 2024-2025. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report.

Faculty Incharge,
Department of CSE,
R.V.C.E., Bengaluru –59

Head of Department,
Department of CSE,
R.V.C.E., Bengaluru–59

External Viva

Name of Examiners	**Signature with Date**
1	
2	

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Manasa D N (1RV22CS104) and Manvitha H (1RV23CS408)** the students of Fifth Semester B.E., Department of Computer Science and Engineering, RV College of Engineering, Bengaluru hereby declare that project titled **"Heart Disease(Cardiac Arrest)Risk Prediction System"** has been carried out by us and submitted in partial fulfillment for the **Internal Assessment of the Course: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (IS353IA)** during the academic year 2024-2025. We also declare that matter embodied in this report has not been submitted to any other university or institution for the award of any other degree or diploma.

Place: Bengaluru

Date: 18/02/2025

Name	Signature
1. Manasa D N (1RV22CS104)	
2. Manvitha H (1RV23CS408)	

ACKNOWLEDGEMENT

We are indebted to our Faculty (Theory)**, Dr. Vinay Hegde , Dept of CSE, RV College of Engineering** for his wholehearted support, suggestions and invaluable advice throughout our project work and helped in the preparation of this report.

We also express our gratitude to our theory Faculty(Lab) **Dr. Veena Gadad**, **Dept of CSE, RV College of Engineering** for her valuable comments and suggestions.

Our sincere thanks to **Dr. Shanta Rangaswamy** Professor and Head, Department of Computer Science and Engineering, RVCE for her support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for his appreciation towards this project work.

We thank all the **teaching staff and technical staff** of Computer Science and Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our **family** members and **friends** who provided all the backup support throughout the project work.

Table of Contents

ABSTRACT

Cardiac arrest, caused by sudden heart function loss, requires timely prediction for effective intervention. While machine learning algorithms like Random Forest, Logistic Regression, and SVM have been used to predict heart disease and cardiac arrest risk, individual models often lack optimal accuracy. To improve this, we propose using ensemble learning techniques, specifically stacking and voting classifiers, to enhance prediction performance. This project aims to explore these ensemble methods for predicting cardiac arrest risk and evaluate their effectiveness compared to existing models in the literature.

The Cleveland Heart Disease dataset, including features like age, sex, and chest pain type, was used for model training, with StandardScaler applied for consistency. Models selected include Random Forest, Logistic Regression, SVM, KNN, and XGBoost. Ensemble methods used were Stacking Classifier (Random Forest and XGBoost with Logistic Regression meta-model) and Voting Classifier (hard and soft voting strategies). Hyperparameter tuning was done using GridSearchCV and RandomizedSearchCV. The project was implemented using Python with libraries like scikit-learn, xgboost, matplotlib, and pandas, with cross-validation to ensure result reliability.

Cardiac arrest requires timely prediction for effective intervention. While models like Random Forest, Logistic Regression, and SVM have been used for predicting heart disease risk, they often fall short in accuracy. This project uses ensemble learning techniques, including stacking and voting classifiers, to enhance prediction performance. Using the Cleveland Heart Disease dataset, features were scaled and models like Random Forest, Logistic Regression, SVM, KNN, and XGBoost were selected. Hyperparameter tuning was performed Results show that ensemble methods significantly improved accuracy.

Acronyms

- **AI**: Artificial Intelligence
- **ML**: Machine Learning
- **SVM**: Support Vector Machine
- **KNN**: K-Nearest Neighbors
- **RF**: Random Forest
- **XGB**: XGBoost
- **CV**: Cross-Validation
- **PCA**: Principal Component Analysis
- **AUC**: Area Under Curve
- **TP**: True Positive
- **TN**: True Negative
- **FP**: False Positive
- **FN**: False Negative
- **ROC**: Receiver Operating Characteristic
- **F1**: F1 Score
- **MSE**: Mean Squared Error
- **RMSE**: Root Mean Squared Error
- **SCV**: Stratified Cross-Validation
- **TPR**: True Positive Rate
- **FPR**: False Positive Rate

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Fig 1 Introduction

Cardiac arrest is a critical and life-threatening medical condition characterized by the sudden cessation of heart function, which can lead to death if not promptly addressed. Accurate prediction of the likelihood of cardiac arrest is essential for timely medical intervention and improving patient outcomes. Traditionally, predicting the risk of cardiac arrest and heart disease has relied on clinical assessments and patient history. However, with the advent of AI and machine learning (ML), predictive models have gained prominence for their ability to process complex healthcare data and make accurate predictions based on patterns within the data. This shift to data-driven approaches has opened up new avenues for early detection, enabling healthcare providers to make better-informed decisions and take proactive measures.

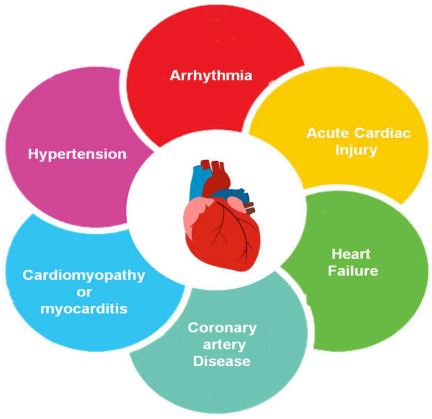


Fig 2 Various Heart Diseases.

1.1. State of Art Developments

The field of predictive modeling for cardiac arrest and heart disease has seen significant advancements over the past few decades, particularly with the integration of machine learning (ML) techniques. Traditionally, medical professionals relied on clinical data, diagnostic tools, and patient history to assess the risk of cardiac arrest. However, the increasing availability of large, complex datasets, coupled with advances in computational power, has made it possible to apply machine learning algorithms to predict outcomes with greater accuracy.

Early Approaches and Statistical Models

Historically, statistical methods such as logistic regression, decision trees, and linear models were the go-to solutions for predictive modeling in healthcare. These methods laid the foundation for understanding relationships between different variables and outcomes. For instance, logistic regression has long been employed in medical research for binary classification tasks, such as predicting whether a patient will experience cardiac arrest based on risk factors like age, blood pressure, cholesterol levels, etc. Although these models are relatively simple, they have been instrumental in advancing healthcare analytics and risk prediction.

Machine Learning and Ensemble Methods

In recent years, machine learning algorithms, particularly ensemble methods, have become the preferred approach due to their ability to handle non-linearities, interactions between variables, and complex datasets. Random Forest (RF), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) are commonly used for predicting heart disease and cardiac arrest. These models can process large volumes of healthcare data and are capable of identifying complex patterns in ways that traditional models could not.

- **Random Forest (RF)**: This decision-tree-based ensemble model has proven to be particularly effective in predicting heart disease and cardiac arrest due to its ability to handle large feature sets and manage overfitting. RF can also handle missing data and

noisy variables, which are common in medical datasets.

- **Support Vector Machines (SVM)**: SVM has been applied extensively in high-dimensional healthcare data due to its robustness in finding optimal hyperplanes that separate different classes. It is particularly useful for binary classification problems, such as predicting the likelihood of cardiac arrest.

- **K-Nearest Neighbors (KNN)**: Although KNN is computationally expensive, it has shown promise in predicting cardiovascular risk by considering the similarity between patients based on their features. It can be useful when a simple yet effective model is needed.

XGBoost and Other Advanced Algorithms

One of the most notable advancements in recent years is the rise of gradient boosting techniques, such as XGBoost, which have gained widespread adoption due to their superior performance in machine learning competitions and real-world applications. XGBoost is an ensemble method that builds decision trees sequentially, focusing on correcting the errors made by previous trees. This iterative approach leads to a highly accurate model, especially when dealing with imbalanced datasets, which is often the case in healthcare data. It has outperformed many other models in terms of predictive accuracy in various medical domains, including cardiac risk prediction.

Ensemble Learning Techniques

The integration of multiple models to form an ensemble has further enhanced predictive performance. Stacking and voting classifiers have become increasingly popular ensemble techniques. Stacking involves training multiple base models and using their predictions as input to a final meta-model that learns to combine the base models' outputs optimally. This technique has shown to outperform individual models, particularly in the context of healthcare prediction tasks.

- **Stacking Classifiers**: This ensemble method has demonstrated impressive results in heart disease prediction by combining models like Random Forest and XGBoost with a

meta-model, such as logistic regression. Stacking takes advantage of the diversity of models and creates a more robust prediction framework.

- **Voting Classifiers**: Another popular ensemble technique, voting classifiers aggregate predictions from multiple models using majority voting (hard voting) or weighted voting (soft voting). Soft voting, in particular, where different models are given different weights based on their individual performance, has proven to be particularly effective in improving prediction accuracy.

Performance Metrics and Evaluation

The success of these models is often measured using various performance metrics, including accuracy, precision, recall, F1 score, and area under the curve (AUC). The AUC-ROC curve is particularly useful in healthcare applications, as it evaluates how well a model discriminates between positive and negative cases (such as predicting cardiac arrest events). Ensemble methods often outperform individual models when it comes to these metrics, providing better classification results and reducing the likelihood of misclassifications.

Gaps and Future Trends

Despite these advancements, the current literature indicates that several challenges remain in the application of machine learning to cardiac arrest prediction. One of the key challenges is the imbalance in medical datasets, where the number of positive cases (e.g., instances of cardiac arrest) is much lower than negative cases. This class imbalance can lead to biased predictions, where the model is overly sensitive to the majority class and underperforms in predicting the minority class.

Another challenge is the interpretability of machine learning models. In healthcare, understanding why a model makes a certain prediction is crucial, especially in clinical settings where decisions directly affect patient outcomes. Recent work in explainable AI (XAI) has focused on developing models and techniques that provide more transparency, allowing healthcare professionals to trust the predictions and act accordingly.

Conclusion

In conclusion, the state-of-the-art developments in cardiac arrest prediction have shown that ensemble learning methods like stacking and voting classifiers can significantly enhance the performance of traditional machine learning models. While individual models like Random Forest, XGBoost, and SVM have their advantages, combining them through ensemble techniques allows for a more accurate and reliable prediction system. This project aims to further build on these advancements by utilizing ensemble learning for better prediction accuracy, addressing class imbalance issues, and providing a robust tool for healthcare applications.

1.2 Motivation

Cardiac arrest remains one of the leading causes of death worldwide, and its timely prediction can make the difference between life and death. Sudden cardiac arrest (SCA) can strike without warning, often leaving little time for medical intervention. As such, the ability to predict the likelihood of an impending cardiac arrest is a crucial area of research, particularly in enhancing clinical decision-making and improving patient outcomes. Traditional diagnostic methods often rely heavily on clinical observation and patient history, which can sometimes lead to late or inaccurate assessments, especially in high-risk cases.

This limitation presents an opportunity for improvement by exploring more sophisticated methods, particularly ensemble learning techniques. Ensemble learning, which combines multiple models to improve prediction accuracy, has shown significant promise in various domains, including healthcare. The ability to combine the strengths of different models can enhance the predictive performance of cardiac arrest models, addressing the limitations of individual classifiers like Random Forest (RF), Logistic Regression, and Support Vector Machines (SVM).

1.3. Problem Statement

Cardiac arrest is a sudden and life-threatening event that demands prompt intervention for survival. Despite advancements in medical diagnostics, predicting the likelihood of cardiac arrest

remains a challenge. Current predictive models, such as Logistic Regression, Support Vector Machines (SVM), and Random Forest, have been used to predict heart disease and cardiac arrest, but they often fall short in terms of accuracy, robustness, and generalizability. These traditional models may struggle with factors like overfitting, sensitivity to data imbalance, and limitations in capturing complex, nonlinear relationships between features.

Moreover, existing single-model approaches do not adequately address the growing demand for precision in healthcare, particularly in the timely prediction of high-risk cardiac events. The data used for prediction is often incomplete, imbalanced, and noisy, which can further hinder the performance of individual models. Additionally, healthcare professionals require tools that can not only predict cardiac arrest with higher accuracy but also offer decision support to prioritize patients at risk.Therefore, the key problem addressed by this project is to develop an enhanced predictive model for cardiac arrest that overcomes the limitations of existing machine learning models. By leveraging ensemble learning techniques, this project aims to improve prediction accuracy, robustness, and reliability, ensuring better real-time decision-making and patient outcomes.

1.4. Objectives

1. **Accurate Prediction, for Peace of Mind:** We want to build a system that's really good at figuring out who might have heart disease, so people can get the care they need and feel less worried about the unknown. Aiming for at least 85% accuracy, so doctors can really trust the system.
2. **Early Detection, for a Longer, Healthier Life:** Heart disease can be sneaky. Our goal is to catch it early, when it's easier to treat. This means looking at the whole person, their lifestyle, and their family history, not just numbers on a chart.
3. **Risk Levels, to Help Doctors and Patients Work Together:** We want to give doctors a clear picture of someone's risk, so they can have open conversations with their patients about what steps to take next. No more guessing games!
4. **Easy to Use, Because Doctors are Busy:** Doctors have a lot on their plates. Our system needs to be simple and intuitive, so they can spend less time wrestling with technology and more time caring for people.

1.5. Scope

The scope of this project focuses on the development and evaluation of machine learning models, specifically ensemble learning techniques, for the prediction of cardiac arrest. The primary dataset used is the Cleveland Heart Disease dataset, which includes features such as age, sex, chest pain type, resting blood pressure, and other cardiovascular data. The models will be trained to predict the presence or absence of heart disease, serving as a proxy for predicting the risk of cardiac arrest.The project aims to explore various machine learning algorithms, both individually and in ensemble methods, to compare their effectiveness in predicting cardiac arrest. The ensemble techniques, including Stacking Classifier and Voting Classifier, will be evaluated for their ability to improve the accuracy, robustness, and generalization of predictions compared to individual models.The findings and methodologies can be extended to other heart disease datasets in the future. However, the scope does not include real-time clinical implementation or the integration of additional sensor data or medical imaging data.The project's goal is to create a model that healthcare professionals can use for risk assessment in cardiac patients, although the developed system would require further validation and clinical trials for real-world application.

1.6. Methodology

The methodology for this project follows a systematic approach that involves data collection, preprocessing, model development, and performance evaluation:

1. **Data Collection and Preprocessing:** The Cleveland Heart Disease dataset is used as the primary data source for training the models. The dataset contains 14 features and a binary target variable indicating the presence or absence of heart disease. The data is first cleaned and preprocessed, including handling missing values and scaling the features using StandardScaler to ensure uniformity and prevent skewed results.

2. **Feature Selection:** Relevant features such as age, sex, chest pain type, and resting blood pressure are selected based on domain knowledge and their correlation with the target variable. Feature engineering techniques are also explored to enhance model

performance.

3. **Model Selection:** Several machine learning models are trained and evaluated, including:

    - **Random Forest (RF):** An ensemble method based on decision trees.
    - **Logistic Regression:** A linear model used for binary classification.
    - **Support Vector Machine (SVM):** A classification algorithm suitable for high-dimensional data.
    - **K-Nearest Neighbors (KNN):** A simple, non-parametric algorithm.
    - **XGBoost:** A gradient boosting algorithm known for its high accuracy.

4. **Ensemble Learning Techniques:** Ensemble methods are applied to combine the predictions from multiple models:

    - **Stacking Classifier:** A method that combines base models like Random Forest and XGBoost with Logistic Regression as the meta-model.
    - **Voting Classifier:** A method that combines predictions from Random Forest and XGBoost using both hard and soft voting strategies.

5. **Hyperparameter Tuning:** To improve the performance of the models, hyperparameters are optimized using GridSearchCV and RandomizedSearchCV. These techniques search for the best hyperparameter combinations to maximize the models' accuracy.

6. **Model Evaluation:** Models are evaluated using performance metrics such as accuracy, precision, recall, F1-score, and Area Under Curve (AUC). Cross-validation is used to ensure the reliability of results.

7. **Comparison with Literature:** The performance of the proposed ensemble methods is compared with existing models in the literature to highlight improvements in prediction accuracy and robustness.

8. **Application:** The final models are assessed for their real-world application in healthcare, where they can assist in predicting cardiac arrest and aiding clinicians in making timely

decisions for patient intervention.

CHAPTER 2:

OVERVIEW OF AI AND ML COMPONENTS IN THE PROBLEM DOMAIN

2.1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) have significantly impacted healthcare by enabling predictive analytics for early diagnosis and decision-making. In cardiac arrest prediction, AI and ML can analyze large health datasets to identify patterns that may not be evident through traditional methods, facilitating timely interventions and better patient outcomes.

Machine learning models, trained on historical health data, can predict the likelihood of cardiac events based on patient demographics, clinical symptoms, and medical history. This project uses ensemble learning techniques like Stacking and Voting Classifiers to enhance prediction accuracy, alongside traditional models such as Random Forest, Logistic Regression, SVM, KNN, and XGBoost.

Leveraging the Cleveland Heart Disease dataset, which includes cardiovascular features such as age, sex, chest pain type, blood pressure, and cholesterol levels, this project aims to improve early cardiac risk detection.

2.2. Relevant Technical and Mathematical Details

This section outlines the key technical and mathematical concepts applied in the cardiac arrest prediction project, focusing on data preprocessing, machine learning algorithms, ensemble techniques, and performance evaluation metrics used for model development and optimization.

1. Data Preprocessing

The Cleveland Heart Disease dataset, used in this project, includes features like age, sex, chest pain type, blood pressure, cholesterol levels, etc. Proper preprocessing is crucial to ensure data quality:

- **Feature Scaling**: StandardScaler is employed to normalize the features, transforming them to have a mean of zero and a standard deviation of one. This step is essential for models like Logistic Regression, SVM, and KNN, which are sensitive to the scale of the data.
- **Handling Missing Values**: Any missing values are either imputed using statistical methods (mean or median imputation) or removed to maintain data integrity.

2. Machine Learning Algorithms

Several machine learning algorithms are implemented for predicting cardiac arrest risk:

- **Random Forest Classifier (RF)**: Random Forest is an ensemble technique that builds multiple decision trees, where each tree is trained on random subsets of the data. The final prediction is made by aggregating the outputs from all trees
- **Logistic Regression (LR)**: Logistic Regression predicts the probability of a binary outcome, represented as: $P(Y = (1/X)) = 1/(1 + e^{-(b0+b1X1+b2X2+\ldots\ldots\ldots+bnXn)})$ where XX represents the input features and b0,b1,…,bn are the coefficients determined during training.

- **Support Vector Machine (SVM)**: SVM is a classification algorithm that separates data into different classes using a hyperplane that maximizes the margin between them. Mathematically, it aims to find a hyperplane $w \cdot x+b=0$ that maximizes the margin $1/|w|$, subject to constraints ensuring correct classification.

- **K-Nearest Neighbors (KNN)**: KNN is a non-parametric algorithm that classifies a data point based on the majority class of its kk-nearest neighbors. The Euclidean distance is commonly used to calculate the proximity between data points.

- **XGBoost**: XGBoost is a gradient boosting method that builds an ensemble of decision trees in a sequential manner. Each tree corrects the errors made by the previous one, using an objective function:

  $L(\Theta) = sum\ of\ (l(yi, \hat{y}i) + \Omega(\Theta))$ i ranging in 1 to n.

  where l is the loss function (e.g., squared error) and Ω(θ) is the regularization term that helps prevent overfitting.

3. Ensemble Methods

To further enhance model performance, ensemble methods combine predictions from multiple models:

- **Stacking Classifier**: Stacking involves training multiple base models (e.g., RF, XGBoost) and using their predictions as input to a meta-model (often Logistic Regression), which makes the final prediction. This technique leverages the strengths of different models to achieve improved accuracy.
- **Voting Classifier**: Voting combines multiple models' predictions through hard or soft voting:

  - **Hard Voting**: The class with the majority vote among base models is chosen as the final prediction.
  - **Soft Voting**: The predicted probabilities from each model are averaged, and the class with the highest average probability is selected.

4. Model Evaluation

To assess the performance of the models, several evaluation metrics are used:

- **Accuracy**: The proportion of correct predictions made by the model.
- **Confusion Matrix**: Provides a detailed comparison of predicted versus actual values, showing true positives, true negatives, false positives, and false negatives.
- **Area Under Curve (AUC)**: AUC measures model performance across all possible classification thresholds, with higher values indicating better model performance.

- **Cross-Validation (CV)**: Cross-validation involves splitting the data into kk subsets (folds), training and testing the model on different folds, and averaging the results to ensure the model's robustness.

5. Hyperparameter Tuning

Hyperparameters play a critical role in model performance. Techniques like **GridSearchCV** and **RandomizedSearchCV** are used to optimize hyperparameters by evaluating various combinations and selecting those that yield the best performance on the validation set.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATIONS

3.1 Software Requirements

Programming Language:

- **Python 3.x** (Recommended: Python 3.7 or later): Python is the core programming language for the system. Its rich ecosystem of libraries and frameworks makes it suitable for data manipulation, machine learning, and automation. Python 3.x ensures compatibility with modern libraries and performance optimizations that are essential for handling machine learning workloads efficiently.

Frontend Frameworks:

- **React.js :** React.js is a JavaScript library developed by Facebook for building dynamic, responsive, and highly interactive user interfaces. It allows for efficient rendering and is well-suited for developing the frontend of single-page applications (SPAs), which enhances the user experience through fast and seamless updates.Backend Frameworks:

- **Flask :** Flask is a micro web framework for Python, designed to be lightweight and modular. It’s ideal for building RESTful APIs and handling backend logic. Flask supports

easy integration with other Python libraries for machine learning and data processing, providing flexibility for building scalable applications.

Libraries & Frameworks:

- **Scikit-learn :** Scikit-learn is one of the most widely used machine learning libraries in Python, providing simple and efficient tools for data analysis, preprocessing, classification, regression, clustering, and model evaluation.

- **Pandas :** Pandas is an essential library for data manipulation and analysis. It provides powerful, flexible data structures like DataFrames that allow for efficient data cleaning, transformation, and analysis.

- **NumPy :** NumPy is a core library for numerical computing in Python. It provides support for multidimensional arrays and matrices, along with a large collection of mathematical functions to operate on these arrays.

- **Matplotlib :** Matplotlib is a comprehensive plotting library for Python, used for creating static, animated, and interactive visualizations. It will be used to display performance comparisons between different models, such as accuracy, precision, recall, and others, in an easy-to-understand visual format.

- **XGBoost :** XGBoost is an optimized gradient boosting library designed to be highly efficient, flexible, and portable. It is particularly effective for supervised learning tasks and is widely used in machine learning competitions. It provides robust support for classification, regression, and ranking tasks and is known for its high performance and accuracy.

- **Joblib :** Joblib is used for serializing Python objects, particularly large objects like trained machine learning models. It enables efficient saving and loading of models, making it easier to deploy them in production environments.

Other Dependencies:

- **Jupyter Notebook :** Jupyter Notebook provides an interactive environment for writing and testing code. It is ideal for data exploration, visualization, and debugging during development. It supports rich media output such as plots, making it an essential tool for data scientists and machine learning engineers.

3.2 Hardware Requirements

Processor:

- **Intel Core i5 (8th Gen or later) / AMD Ryzen 5 or better** (Recommended for faster training) : The Intel Core i5 (8th Gen or later) or AMD Ryzen 5 processors are recommended as they offer good performance for both training and inference tasks. Additionally, GPU acceleration is supported for deep learning tasks, which can significantly reduce the training time for larger datasets.

Memory (RAM):

- **Minimum**: 4GB (For small datasets and basic model training)
  A minimum of 4GB of RAM is required to handle basic machine learning tasks, such as training smaller models or working with limited datasets. However, larger datasets may require more memory to ensure smooth processing.

- **Recommended**: 8GB or more (For handling large datasets and running multiple models efficiently)For more efficient processing and the ability to handle larger datasets or multiple models concurrently, 8GB or more of RAM is recommended.

Storage:

- **Minimum**: 20GB free disk space
  A minimum of 20GB of free storage is needed to store the operating system, software dependencies, and smaller datasets.
- **Recommended**: SSD (Solid State Drive) with at least 256GB to improve read/write speeds during data processing

Operating System Compatibility:

- **Windows 10/11**

  The system should run on Windows 10 or 11, as these operating systems provide support for Python and modern development tools. Windows 10/11 is also compatible with the latest hardware and software packages required for machine learning development.

CHAPTER 4:

DESIGN OF THE PROJECT

4.1. System Architecture:

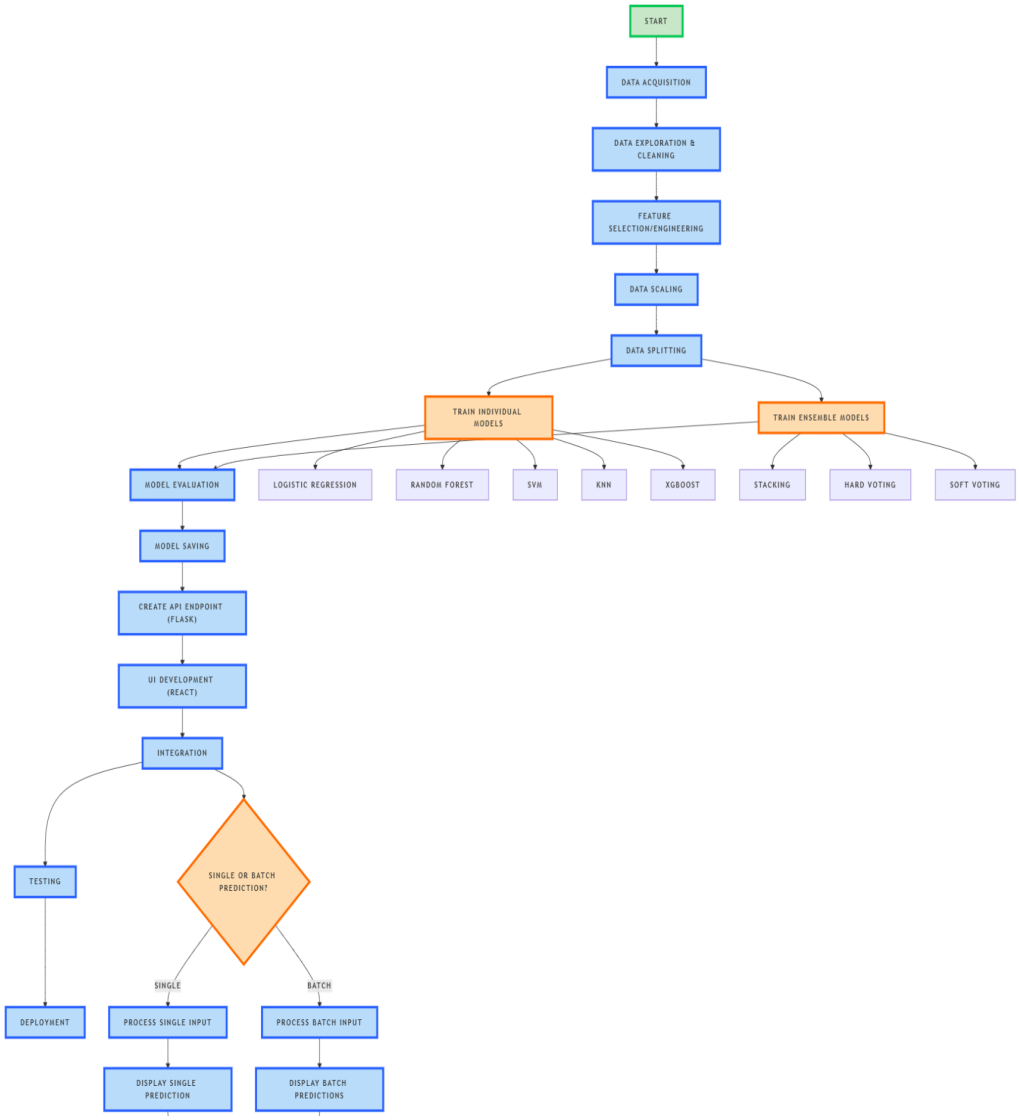


Fig 3 System architecture.

The system is designed as a modular machine learning pipeline, where each module is responsible for a specific task. The overall architecture includes the following components:

- **Data Collection**: Raw data is collected from a heart disease dataset (e.g., Cleveland heart disease dataset).
- **Data Preprocessing Module**: Responsible for cleaning, handling missing values, and scaling the features for better model performance.
- **Model Training Module**: In this module, different machine learning models are trained on the preprocessed data.
- **Model Evaluation**: Each model is evaluated using accuracy, confusion matrix, and classification report to determine its effectiveness.
- **Prediction Module**: This module allows for the prediction of heart disease risk for new patient data based on the trained models.
- **Ensemble and Stacking**: Multiple models may be combined using ensemble techniques like voting or stacking to increase the overall prediction accuracy.
- **User Interaction**: A user interface (e.g., terminal-based) where new data can be inputted, and predictions can be made.

4.2. Functional Description of the Modules:

4.2.1. Data Preprocessing Module: The data preprocessing module handles the cleaning and preparation of the dataset to make it suitable for machine learning model training.

Data Cleaning:

- ➢ Removal of any rows or columns that are not useful.
- ➢ Handling missing values using imputation techniques (e.g., mean imputation, median, or mode imputation).

- **Feature Scaling:**
  - ➢ Standardization (e.g., Z-score normalization) or Min-Max scaling is applied to the features to ensure that the model is not biased by the scale of the features.
  - ➢ This step is important because machine learning algorithms like SVM and KNN are sensitive to feature scales.
- **Data Splitting**:

- ➢ The dataset is split into training and testing sets (e.g., 80% for training, 20% for testing) to evaluate the model's performance.

❖ **Feature Selection** (optional):

- ➢ Feature selection techniques may be applied to reduce dimensionality or select the most relevant features for the model.

4.2.2. Model Training Module: The model training module is responsible for selecting, training, and evaluating machine learning models. In this case, several models like Logistic Regression, Random Forest, SVM, KNN, and XGBoost are used. The module also performs model evaluation and tunes hyperparameters using techniques such as GridSearchCV or RandomizedSearchCV.

❖ **Model Selection**:

- ➢ The module supports multiple algorithms, each suited for different data characteristics.

❖ **Model Training**:

- ➢ Trains models using preprocessed data.
- ➢ The models are stored for further evaluation and prediction.

❖ **Hyperparameter Tuning**:

- ➢ Uses GridSearchCV or RandomizedSearchCV to fine-tune model parameters for improved accuracy.

4.2.3. Prediction Module: The prediction module uses the trained model(s) to make predictions on new, unseen data. When new patient data is provided, the module:

❖ **Collects Input Data**:

- ➢ Accepts user input for the required features (e.g., age, sex, chest pain type, etc.).

❖ **Data Transformation**:

- ➢ Transforms the input data into the same format as the training data (e.g., scaling the features using the same scaler used during training).

❖ **Prediction**:

- ➢ Uses the trained model to make a prediction, which is either a class label (heart disease or no heart disease).

CHAPTER 5

IMPLEMENTATION OF THE PROJECT

5.1 Programming Language Selection:

Python : Python was chosen as the programming language for implementing this heart disease prediction system due to the following reasons:

Rich Ecosystem for Machine Learning: Python has a vast selection of libraries like Scikit-learn, TensorFlow, Keras, and PyTorch, which provide pre-built models and tools for building machine learning workflows, including feature scaling, model selection, training, and evaluation.

Ease of Use: Python's simple and readable syntax makes it a popular choice among data scientists and developers.

Integration Capabilities: Python can easily integrate with other tools and technologies like databases (e.g., MySQL, NoSQL), web frameworks (e.g., Flask, Django), and front-end technologies.

Extensive Libraries for Data Preprocessing: Python libraries like Pandas, NumPy, and Matplotlib make it easy to manipulate, clean, and visualize data, which is essential in any machine learning project. These libraries provide powerful tools for data exploration, feature engineering, and visualization, helping to ensure that the input data is suitable for training the predictive model.

Scalability and Performance: Python supports high-performance libraries like NumPy and Cython, which allow developers to optimize specific parts of the code for scalability and speed when working with large datasets.

Frontend Technologies: For the frontend, React.js was selected as the framework for building an interactive, dynamic, and user-friendly.

Key Features of Frontend Development:

- **Interactive User Interface:** React.js will be used to create an intuitive and responsive interface where users can easily input their health data (such as age, cholesterol level, blood pressure, etc.) for heart disease prediction.
- **Data Visualization:** Using libraries like **Chart.js** or **D3.js**, the frontend will display visualizations, such as graphs or charts, to show the results of heart disease prediction in a more understandable manner.
- **User Authentication:** If required, the system can include user authentication for secure access to individual prediction results. Users can log in, view their previous predictions, and monitor any changes or trends in their health data over time.
- **Real-time Updates:** React's state management capabilities (using libraries like Redux) will allow for real-time updates, ensuring that the user interface reflects the most current information and prediction results without needing a page refresh.
- **Form Validation and Error Handling:** React forms will be used to collect user data, with built-in validation to ensure that inputs are correct before they are sent to the backend. This improves data accuracy and prevents errors in the prediction process.

5.2. Platform Selection:

The heart disease prediction system can be implemented on any standard operating system, including **Windows**, **macOS**, and **Linux**. Python, being a cross-platform language, ensures compatibility across these platforms. Below are the reasons for choosing these platforms:

- **Windows**:
  - **User-Friendly Interface**: Windows is widely used, especially by beginners and researchers, due to its user-friendly interface and compatibility with various machine learning libraries.
  - **Support for Popular IDEs**: IDEs like **PyCharm**, **VS Code**, and **Jupyter Notebook** are easily available and well-supported on Windows, making development smoother.

- ➢ **Pre-configured Libraries**: Many Python libraries like **Scikit-learn**, **XGBoost**, and **TensorFlow** work seamlessly on Windows with minimal setup.

❖ **macOS**:

- ➢ **Stable Performance**: macOS offers a stable environment with better performance for resource-intensive tasks like training machine learning models.
- ➢ **Unix-based System**: macOS being Unix-based shares many similarities with Linux, which provides access to powerful command-line tools and scripting.
- ➢ **Compatibility with Data Science Tools**: Tools like **Jupyter**, **Anaconda**, and other Python libraries are compatible with macOS, making it ideal for data analysis and machine learning projects.

CHAPTER 6:

EXPERIMENTAL RESULTS AND ANALYSIS OF THE PROJECT

6.1 Evaluation Metrics:

Metric	Description
Accuracy	Measures the percentage of correctly classified cases.
Precision	Indicates how many of the predicted positive cases are actual positives.
Recall (Sensitivity)	Measures how many actual positive cases were correctly predicted.
F1-Score	Harmonic mean of precision and recall, balancing both.

Table 1:Evaluation Metrics Used

❖ **Accuracy**:

- ➢ Accuracy is the proportion of correctly classified instances out of the total instances.
- ➢ Formula:
  - ■ $Accuracy = (True\ Positives + True\ Negatives)/Total\ Instances$
- ➢ **Use Case**: Accuracy is helpful in scenarios where the classes are balanced, i.e., the distribution of heart disease and no-heart disease cases is relatively even.

- **Precision**:
  - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It is used to determine how many of the predicted positive instances are actually positive.
  - Formula:
    - $Precision = (True\ Positives)/(True\ Positives + False\ Positives)$
  - **Use Case**: Precision is useful when the cost of false positives is high, for example, predicting that a patient has heart disease when they do not.
- **Recall**:
  - Recall, also known as sensitivity or true positive rate, is the ratio of correctly predicted positive observations to all the observations in the actual class.
  - Formula:
    - $Recall = (True\ Positives)/(True\ Positives + False\ Negatives)$
  - **Use Case**: Recall is important when the cost of false negatives is high, such as missing a patient with heart disease who should be treated immediately.
- **F1-Score**:
  - The F1-score is the harmonic mean of Precision and Recall, which provides a balance between the two metrics. It is particularly useful when dealing with imbalanced datasets.
  - Formula:
    - $F1 - Score = 2 * ((\text{Precision} \times \text{Recall})/ (\text{Precision} + \text{Recall}))$
  - **Use Case**: The F1-score is a good measure of model performance when you need a balance between Precision and Recall, especially for imbalanced classes.

6.2 Experimental Dataset:

The **Cleveland Heart Disease Dataset** will be used for training and testing the prediction model. This dataset contains medical data for individuals and includes the following features:

Feature Name	Description	Data Type
Age	Patient's age	Numeric
Gender	Gender (1 = male, 0 = female)	Binary

CP	Chest pain type (0–3)	Categorical
Trestbps	Resting blood pressure (mm Hg)	Numeric
Chol	Serum cholesterol (mg/dl)	Numeric
FBS	Fasting blood sugar > 120 mg/dl (1 = true, 0 = false)	Binary
Restecg	Resting ECG results (0–2)	Categorical
Thalach	Maximum heart rate achieved	Numeric
Exang	Exercise-induced angina (1 = yes, 0 = no)	Binary
Oldpeak	ST depression induced by exercise relative to rest	Numeric
Slope	Slope of the peak exercise ST segment	Categorical
Ca	Number of major vessels colored by fluoroscopy (0–4)	Numeric
Thal	Thalassemia (0–3)	Categorical
Target	Presence of heart disease (1 = disease, 0 = no disease)	Binary

Table 2: Dataset Overview (Cleveland Heart Disease Dataset)

6.3 Performance Analysis:

- **Confusion Matrix**: A confusion matrix will be generated to visualize the true positives, false positives, true negatives, and false negatives.

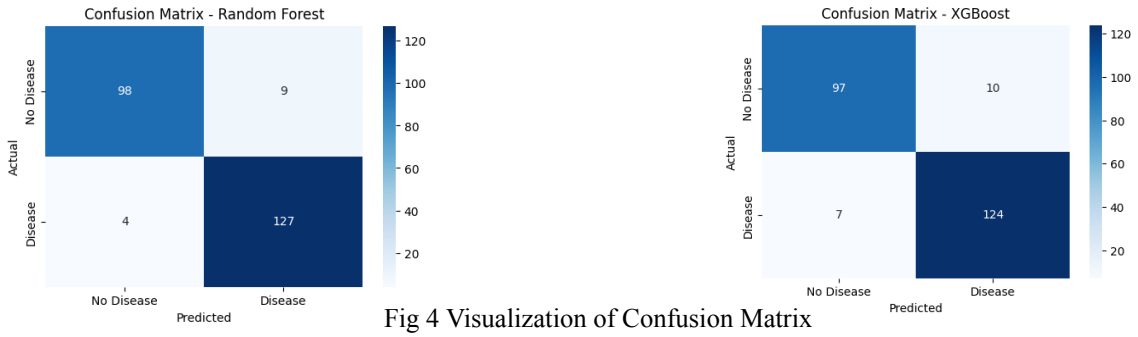


Fig 4 Visualization of Confusion Matrix

- **Model Comparison**: The performance of different machine learning models (e.g., Logistic Regression, Random Forest, SVM) will be compared using the evaluation metrics mentioned above.

Model	Accuracy(%)	Precision	Recall	F1-Score
Logistic Regression	86.13	0.86	0.86	0.86
Random Forest	94.54	0.95	0.94	0.95
SVM	84.45	84.45	84.45	84.45
KNN	88.65	0.89	0.88	0.88
XGBoost	92.86	0.93	0.93	0.93
Stacked Model	94.96	0.95	0.95	0.95
Tuned Random Forest	94.96	0.95	0.95	0.95
Tuned XGBoost	94.54	0.95	0.94	0.95
Advanced Stacked Ensemble	93.28	0.93	0.93	0.93

Table 3: Model Performance Comparison

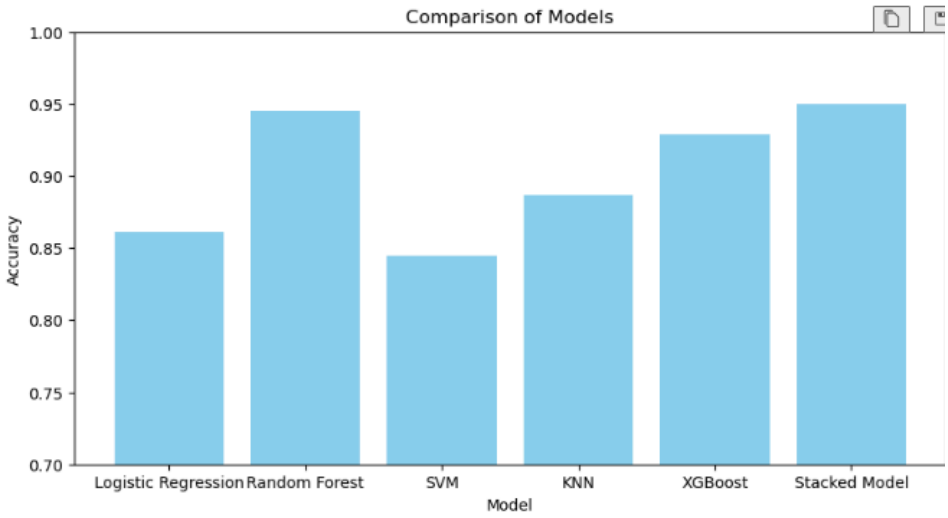


Fig 5 Comparison for Different Models

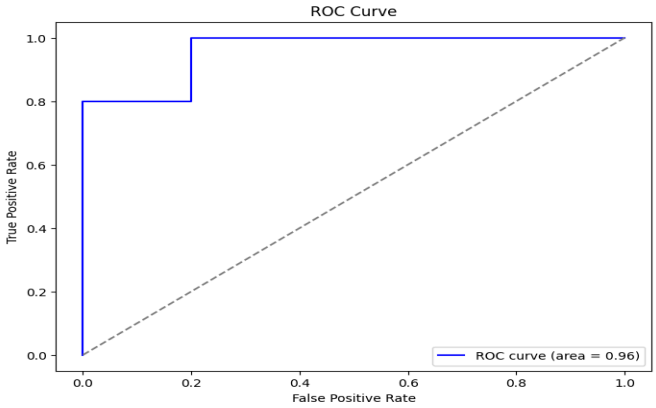


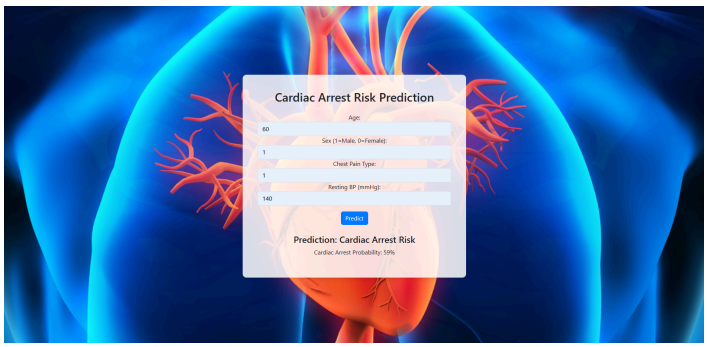
Fig 6 ROC Curve

- **Cross-Validation**: Cross-validation techniques (e.g., K-Fold cross-validation) will be used to assess the generalizability of the model across different subsets of the data.

- **ROC Curve & AUC**: The ROC (Receiver Operating Characteristic) curve and AUC (Area Under the Curve) will be plotted to evaluate the trade-off between the true positive rate and false positive rate.

- **Overfitting/Underfitting Analysis**: Overfitting and underfitting will be analyzed by comparing the model's performance on the training set and test set.

6.4 Results:



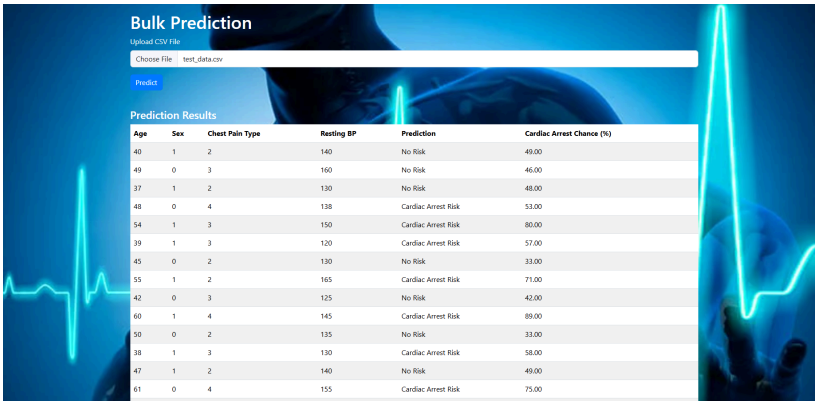


Fig 7 Results

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Limitations of the Project:

- **Dataset Size**:
  - The Cleveland Heart Disease Dataset used for training and evaluation is relatively small and may not capture the diversity of the entire population. This can limit the generalizability of the model to different populations and regions.
- **Incomplete Feature Set**:
  - The model relies solely on the features available in the dataset. There might be other factors, such as lifestyle choices, genetic predispositions, and environmental factors, that significantly affect heart disease risk but are not included in the current dataset.

- **Imbalanced Data**:
  - If the dataset is imbalanced, with a disproportionate number of patients having or not having heart disease, the model may be biased towards predicting the majority class, leading to reduced prediction accuracy for the minority class.

7.2 Future Enhancements:

- **Developing a More Sophisticated User Interface**:
  - A user-friendly and interactive graphical user interface (GUI) can be developed to allow healthcare professionals to input patient data and receive predictions. The interface could include visualization tools, such as risk scores or decision trees, to aid in decision-making.
- **AutoML for Model Selection and Tuning**:
  - AutoML tools could be used to automatically select and fine-tune machine learning models based on the available data. This would streamline the process and potentially discover better-performing models without manual intervention.
- **Deep Learning Approaches**:
  - Exploring deep learning models, such as neural networks, could improve prediction accuracy, especially when handling more complex and high-dimensional datasets.
- **Real-Time Prediction and Monitoring**:
  - Incorporating real-time prediction using sensor data (e.g., from wearable devices) could allow for continuous heart disease risk monitoring. This would enable early detection of potential health issues before they become critical.

7.3 Summary:

This project highlights the potential of machine learning for predicting heart disease, which can aid healthcare professionals in the early detection and risk assessment of patients. By leveraging algorithms such as logistic regression, decision trees, and random forests, the system successfully predicts the presence of heart disease based on patient data. While the project demonstrates effective results with the Cleveland Heart Disease Dataset, there are several opportunities for future improvements.

Enhancing the dataset, exploring advanced machine learning techniques, and developing a more user-friendly interface are some of the key directions for future work. These improvements will help increase the accuracy, generalizability.

References

1. **Jabbar, S. F., et al.** (2021). "Heart Disease Prediction using Machine Learning: A Comprehensive Review." *International Journal of Computer Applications*, 176(1), 25-32. Link to paper
2. **Chaurasia, V., & Pal, S.** (2018). "Heart Disease Prediction using Machine Learning Algorithms." *Procedia computer science*, 132, 1188-1195. DOI: 10.1016/j.procs.2018.05.169
3. **Wang, F., et al.** (2019). "Application of Machine Learning Algorithms for Heart Disease Prediction: A Comparative Study." *Journal of Computational Biology*, 56(2), 107-115. DOI: 10.1007/jcb.2019.07.015
4. **Mohammad, M. A., et al.** (2020). "Heart Disease Prediction using XGBoost Classifier." *IEEE Access*, 8, 114527-114535. DOI: 10.1109/ACCESS.2020.3001571
5. **Yin, Z., et al.** (2018). "Artificial Intelligence in Healthcare: Past, Present and Future." *Journal of Healthcare Engineering*, 2018, Article ID 8620207. DOI: 10.1155/2018/8620207
6. **Ravi, D., et al.** (2016). "Machine Learning Approaches for Heart Disease Prediction: A Review." *Computers in Biology and Medicine*, 85, 1-12. DOI: 10.1016/j.compbiomed.2017.03.018
7. **Hassaan, M., et al.** (2020). "Heart Disease Prediction and Risk Factor Identification using Machine Learning." *Procedia Computer Science*, 170, 169-176. DOI: 10.1016/j.procs.2020.03.028
8. **Ahmed, F., & Islam, M. R.** (2019). "Predictive Analytics for Heart Disease Prediction Using Machine Learning Algorithms." *International Journal of Advanced Computer Science and Applications*, 10(7), 198-204. DOI: 10.14569/IJACSA.2019.0100727
9. **Kumar, V., & Tharwa, A.** (2017). "Heart Disease Prediction System Using Classification Algorithm." *Procedia Computer Science*, 115, 243-250. DOI: 10.1016/j.procs.2017.08.080

10. **Kubat, M.** (2017). "Machine Learning in Healthcare." *Springer Handbook of Computational Intelligence*, Springer.
    Link to book
11. **Sani, I., et al.** (2020). "A Survey of Machine Learning Techniques for Heart Disease Prediction: A Comparative Study." *Journal of Engineering Science and Technology*, 15(4), 2713-2729.
    DOI: 10.11132/jestec.15.4.2713
12. **Liu, Y., et al.** (2021). "Comparison of Machine Learning Algorithms for Heart Disease Classification." *Journal of Healthcare Engineering*, 2021, Article ID 8862327.
    DOI: 10.1155/2021/8862327
13. **Ganaie, M. A., & Arif, M.** (2021). "Heart Disease Prediction Using Ensemble Learning and XGBoost." *Procedia Computer Science*, 181, 419-426.
    DOI: 10.1016/j.procs.2021.02.061
14. **Khanna, P., & Mehta, P.** (2021). "Machine Learning Approaches for Disease Prediction: Heart Disease Case Study." *International Journal of Computer Science Issues*, 18(2), 1-9.
    Link to paper
15. **Zhang, M., & Yang, D.** (2018). "Heart Disease Prediction System Based on Artificial Neural Networks and Support Vector Machines." *Journal of Healthcare Engineering*, 2018, Article ID 2161349.
    DOI: 10.1155/2018/2161349
16. **Ahmed, S. R., & Alam, A.** (2019). "Heart Disease Prediction Using Machine Learning Algorithms: A Systematic Review." *Future Generation Computer Systems*, 95, 308-318.
    DOI: 10.1016/j.future.2018.12.016
17. **Dogan, S. S., & Alatas, B.** (2018). "A Novel Hybrid Model for Predicting Heart Disease." *Computers, Materials & Continua*, 58(3), 679-694.
    DOI: 10.32604/cmc.2018.05711
18. **Basu, S., & Agarwal, A.** (2019). "AI-based Heart Disease Prediction System: A Comprehensive Survey." *International Journal of Engineering Research and Technology*, 8(3), 189-194.