VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

Project work Report (21ISP76)
on

"CARDIOVASCULAR DISEASE PREDICTION SYSTEM USING ENSEMBLE TECHNIQUE"

Submitted in partial fulfillment of the requirements for the award of a degree of

Bachelor of Engineering
in

INFORMATION SCIENCE AND ENGINEERING

Submitted by

ABHISHEK G	**1BI21IS114**
THEJU T S	**1BI21IS124**
BHARGAVA K R	**1BI21IS125**
MANAS CHAGI	**1BI21IS051**

Under the Guidance of
Dr. Hema Jagadish
Associate Professor
Dept. of ISE, BIT

BANGALORE INSTITUTE OF TECHNOLOGY
K. R. Road, V. V. Pura, Bengaluru –560004
Department of Information Science Engineering

2024-25

BANGALORE INSTITUTE OF TECHNOLOGY

K.R. Road, V.V. Pura, Bengaluru-560004

Department of Information Science & Engineering

CERTIFICATE

Certified that the Project Work entitled **"CARDIOVASCULAR DISEASE PREDICTION SYSTEM USING ENSEMBLE TECHNIQUE"** carried out by **ABHISHEK G (1BI21IS114), THEJU T S (1BI21IS124), BHARGAVA K R (1BI21IS125), MANAS CHAGI (1BI21IS051)** a bonafide student of Department of ISE, BIT in partial fulfillment of the requirements for the **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi during the academic year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements for the said degree.

Guide:	**HOD:**	**Principal:**
Dr. Hema Jagadish	**Dr. Asha T**	**Dr. Aswath M.U**
Associate Professor	Professor and HOD	Principal
Department of ISE	Department of ISE	BIT, Bangalore
BIT, Bangalore	BIT, Bangalore	

Name of the Examiners	**Signature with Date**
1._____	_____
2._____	_____

DECLARATION

We, **ABHISHEK G (1BI21IS114), THEJU T S (1BI21IS124), BHARGAVA K R (1BI21IS125) MANAS CHAGI (1BI21IS051),** students of VII Semester B.E., in Information Science and Engineering, Bangalore Institute of Technology hereby declare that the Project entitled **"CARDIOVASCULAR DISEASE PREDICTION SYSTEM USING ENSEMBLE TECHNIQUE"** has been carried out by us and submitted in partial fulfilment of the requirements for the VII Semester degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi during academic year 2024-25.

Date:24/12/24

Place: Bangalore

Abhishek G	1BI21IS114
Theju T S	1BI21IS124
Bhargava K R	1BI21IS125
Manas Chagi	1BI21IS051

ACKNOWLEDGEMENT

We express our gratitude with great pleasure to the **Bangalore Institute of Technology**, Bangalore, for providing us the opportunity to fulfill our cherished desire to attain our goal.

We extend our sincere thanks to our principal, **Dr. Aswath M.U**, for his encouragement in all aspects to complete the project.

We would like to thank **Dr. Asha T**, Professor and Head of the Department of ISE, BIT, for her unwavering support and encouragement.

We immensely thank **Dr. Hema Jagadish**, Associate Professor, Dept. of ISE, for supporting and guiding us in our preparations for the project.

Finally, we thank everyone who has helped us directly or indirectly in the completion of the project.

ABHISHEK G	**1BI21IS114**
THEJU T S	**1BI21IS124**
BHARGAVA K R	**1BI21IS125**
MANAS CHAGI	**1BI21IS051**

ABSTRACT

The project focuses on predicting cardiovascular disease using ensemble machine learning technique and ECG image processing. The system enhances ECG images by converting them to grayscale, reducing noise with Gaussian smoothing, and extracting waveforms. Principal Component Analysis (PCA) helps reduce dimensionality while preserving key features. A VotingClassifier combines SVM, kNN, Random Forest, Gaussian Naive Bayes, and Logistic Regression, using a soft voting strategy to improve accuracy. The project is Built with scikit- learn and Streamlit, the system provides an easy-to-use interface for medical professionals to upload ECG images and receive predictions, highlighting the potential of AI for non- invasive heart disease diagnosis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Cardiovascular disease is one of the leading causes of death globally, highlighting the need for early detection and accurate diagnosis to improve patient outcomes. Traditional diagnostic methods, particularly those based on Electrocardiogram (ECG) signals, often require manual analysis, which can be time-consuming and prone to human error. The project Cardiovascular Disease Prediction Using Ensemble seeks to automate this process by utilizing advanced image processing and machine learning techniques to predict cardiovascular diseases from ECG images. By employing an ensemble technique, the system combines the strengths of multiple models to enhance prediction accuracy and reliability. This report outlines the methodologies, data processing steps, and model implementation involved in creating a robust ECG-based prediction system for cardiovascular disease.

1.1 General Overview

Cardiovascular disease is a broad term for conditions affecting the heart and blood vessels, and it remains one of the most common causes of death globally. Early diagnosis of this is critical for reducing the risks of severe complications, such as heart attacks and strokes. Traditional diagnostic methods rely heavily on clinical expertise and manual analysis of ECG signals, which can be time-consuming and prone to subjective errors. As a result, there is a growing need for automated solutions that can analyze ECG data with high accuracy and efficiency.

The Cardiovascular Disease Prediction Using Ensemble project aims to address these challenges by developing a robust system for the prediction of cardiovascular diseases based on ECG images. The system combines image processing techniques and machine learning algorithms to preprocess ECG data and predict the likelihood of CVD. By utilizing an ensemble technique, which combines multiple models to make predictions, the approach aims to improve prediction accuracy and minimize errors that may arise from using a single model.

The process begins by converting ECG images into a format suitable for analysis, followed by preprocessing steps that enhance the quality of the data. The system then extracts relevant features from the ECG signals using techniques such as contouring and signal scaling. These

features are reduced to lower dimensions using Principal Component Analysis (PCA), and finally, a trained ensemble model is used to predict the presence of cardiovascular diseases. The ensemble approach ensures that the system can leverage the strengths of multiple models, leading to more reliable and accurate predictions. This approach not only enhances diagnostic precision but also provides a fast, scalable, and automated method for cardiovascular disease prediction, which can be invaluable in clinical settings.

1.2 Problem Definition

Cardiovascular diseases are among the leading causes of death worldwide, with millions of people affected each year. Early detection of the disease is critical for preventing severe health outcomes, such as heart attacks, strokes, and other life-threatening conditions. However, the current methods of diagnosing cardiovascular diseases are often reliant on manual interpretation of medical tests like ECG (Electrocardiogram) signals, which require significant time, expertise, and are subject to human error. This makes the early and accurate diagnosis of the disease a challenge, especially in resource-constrained settings or where expert clinicians are not readily available. Furthermore, the growing volume of ECG data and the complexity of interpreting these signals pose an additional hurdle in the efficient and accurate detection of heart-related diseases. Traditional methods of analyzing ECGs are slow and may not be able to detect subtle changes in the signals that indicate the early onset of cardiovascular issues. As a result, there is an urgent need for automated, scalable, and accurate systems that can efficiently analyze ECG data and predict the likelihood of disease.

The main problem this project addresses is the lack of an efficient, reliable, and automated system to predict cardiovascular diseases from ECG signals. The goal is to design a model that leverages image processing techniques and machine learning algorithms, particularly ensemble methods, to enhance the accuracy of predictions. This solution aims to help in early detection, enabling better clinical decision-making and timely interventions, while reducing the dependency on manual analysis of ECG signals.

1.3 Objectives

The primary objective of the "Cardiovascular Disease Prediction Using Ensemble Techniques" project is to develop an efficient and automated system for predicting cardiovascular diseases from ECG images. To achieve this, several specific objectives need to be met:

1. ECG Image Preprocessing: To enhance the quality of ECG images, apply preprocessing techniques such as grayscale conversion, Gaussian smoothing, and thresholding to remove noise and improve the clarity of important features in the ECG signals.
2. Lead Division and Signal Extraction: Extract individual ECG leads (12 standard leads and 1 long lead) from the ECG images to isolate the distinct heart electrical signals. This step is crucial for focusing on different regions of the ECG and preparing them for further analysis.
3. Feature Extraction and Signal Transformation: Extract meaningful features from the ECG signals by identifying key signal characteristics. This will involve techniques such as contour extraction, dimensionality reduction using Principal Component Analysis (PCA), and scaling for normalization.
4. Ensemble Learning Model Development: Develop a robust classification model using ensemble techniques. By combining multiple classifiers, such as Random Forest, Gradient Boosting, and Support Vector Machines, the goal is to enhance prediction accuracy and provide a more reliable classification of the ECG signals.
5. Model Evaluation and Performance Assessment: Evaluate the model using metrics such as accuracy, precision, recall, and F1-score to ensure that the system can correctly classify ECG signals into different categories, such as normal heart rhythm, abnormal heartbeats, or indications of myocardial infarction.
6. Prediction for Early Diagnosis: To provide an accurate and timely prediction of cardiovascular diseases, enabling early intervention, and supporting clinical decision-making for healthcare professionals.

By achieving these objectives, the project will contribute to the field of cardiovascular disease prediction and help in the development of a reliable, automated tool that can assist clinicians in diagnosing heart diseases more efficiently.

1.4 Scope

The project focuses on developing a machine learning-based system for predicting cardiovascular diseases using ECG images. The scope includes processing ECG images through grayscale conversion, Gaussian filtering, and binary thresholding to enhance signal quality. It involves extracting 12 standard leads and one long lead from the ECG images, followed by feature extraction, transformation, and dimensionality reduction using PCA. Ensemble learning

techniques, including Random Forest, Gradient Boosting, and SVM, will be employed to improve classification accuracy and robustness. The system will predict heart conditions like abnormal rhythms and myocardial infarction, providing early diagnosis support for healthcare professionals. Model evaluation will rely on metrics such as accuracy, precision, recall, and F1-score. The project is limited to digital ECG image data and a predefined dataset, requiring further training for broader real-world applications.

1.5 Motivation

Cardiovascular diseases are a leading cause of mortality worldwide, necessitating early detection and accurate diagnosis to improve patient outcomes. Traditional diagnostic methods, such as manual ECG interpretation, are often time-consuming and error-prone. This project aims to leverage machine learning, specifically ensemble techniques, to analyze ECG images for automated and accurate prediction. By combining multiple models, ensemble methods enhance prediction performance, offering a faster, more reliable, and cost-effective solution. The ultimate goal is to assist healthcare professionals in diagnosing heart diseases efficiently, reduce their workload, and improve heart health management, particularly in resource-limited settings.

1.6 Methodologies

The methodologies of the project include:

1. ECG Image Collection and Preprocessing

a. Collect ECG images in formats such as PNG, JPG, or JPEG.

b. Convert images to grayscale to simplify analysis and reduce complexity.

c. Apply Gaussian filtering to reduce noise and smooth the image.

d. Use Otsu's method for thresholding to separate signals from the background.

2. Dividing ECG Leads

a. Divide ECG images into 13 individual leads (12 standard leads and 1 long lead).

b. Treat each lead as a separate region of interest (ROI) for analysis.

3. Signal Extraction and Feature Scaling

a. Enhance ECG signals using contour detection to extract features.

b. Normalize signal values with Min-Max Scaling for machine learning models.

4. Dimensionality Reduction with PCA

a. Use Principal Component Analysis (PCA) to reduce dimensionality.

b. Retain the most significant features while removing redundant information.

5. Ensemble Learning for Classification

a. Employ Logistic Regression, KNN, Decision Trees, Random Forest, and Naive Bayes.

b. Combine predictions of individual models for a robust final diagnosis.

6. Model Training and Evaluation

a. Train models on labeled ECG data and apply cross-validation.

b. Evaluate performance using metrics like accuracy, precision, recall, and F1-score.

c. Select the best-performing model based on evaluation results.

7. Prediction and Results

a. Use the trained model to predict cardiovascular diseases, such as myocardial infarction.

b. Provide a diagnostic output to assist healthcare professionals in decision-making

1.7 Project Outcomes

The outcomes of the project are:

1. Accurate Prediction of Cardiovascular Diseases

- Developed a model capable of predicting cardiovascular diseases from ECG images.
- Achieved improved accuracy by combining multiple machine learning algorithms in an ensemble approach.
- Detected conditions such as myocardial infarction, arrhythmias, and other heart abnormalities.

2. Robust Data Processing Pipeline

- Created a comprehensive pipeline for ECG image preprocessing, including grayscale conversion, noise reduction, and thresholding.

- Segmented ECG leads and extracted relevant signal features for machine learning analysis.
- Ensured the quality of input data to optimize feature extraction for model training.

3. Effective Use of Dimensionality Reduction

- Applied Principal Component Analysis (PCA) to reduce dimensionality while preserving key features.
- Reduced overfitting and improved model generalization to new data.

4. Real-Time Predictive System

- Demonstrated the feasibility of a real-time system for predicting cardiovascular diseases in clinical settings.
- Enabled healthcare professionals to upload ECG images and receive predictions for early diagnosis and intervention.

5. Integration of Ensemble Learning for Improved Performance

- Integrated multiple machine learning models (logistic regression, decision trees, KNN, random forests, Naive Bayes) into an ensemble framework.
- Mitigated individual model weaknesses, improving overall prediction performance.

6. Evaluation and Validation of the Model

- Implemented cross-validation and performance metrics (accuracy, precision, recall, F1-score) to assess model reliability.
- Achieved high sensitivity and specificity in detecting cardiovascular diseases.

7. Potential for Future Enhancements

- Expanded future possibilities include incorporating more diverse ECG data and real-time ECG streams.
- Potential for integration with wearable health devices or hospital-based diagnostic tools for improved healthcare delivery.

1.8 Limitations

The limitations of the project are:

1.8.1 Data Limitations

- Limited diversity in the dataset may affect model generalization to various ECG patterns.

- Potential imbalance in cardiovascular disease representation can introduce biases in predictions.

1.8.2 Dependency on High-Quality ECG Images

- Model accuracy is influenced by the quality of ECG images; poor-quality data may hinder feature extraction.
- Noise, distortions, and improper scaling in real-world ECG images may degrade system

1.8.3 Limited Scope of ECG Features

- The system only uses visual ECG features and does not consider other relevant clinical factors.
- Excluding patient history, comorbidities, and genetic data may limit predictive power in complex cases.

1.8.4 Model Complexity and Training Time

- The ensemble approach increases model complexity, leading to longer training times.
- High computational requirements may present challenges in resource-constrained or real-time clinical settings.

1.8.5 Need for Continuous Validation and Testing

- The model requires ongoing validation with new ECG data to maintain reliability.
- Periodic updates and retraining are essential to keep the model current with evolving disease markers.

1.8.6 Ethical and Legal Concerns

The Cardiovascular Disease Prediction System, while designed to be a powerful tool for early detection and risk assessment, has certain limitations. One primary limitation is its reliance on the quality and completeness of input data. Inaccurate, noisy, or incomplete ECG data can adversely affect the system's predictions. The system also depends on pre-trained machine learning models, which may not generalize well to diverse populations or rare conditions unless regularly updated with new and diverse datasets which could enhance longitudinal tracking and personalized predictions. or high user concurrency might require additional infrastructure.

CHAPTER 2

LITERATURE SURVEY

A literature survey involves reviewing existing research and studies related to a specific topic or field. It provides a comprehensive understanding of the current knowledge, methodologies, and advancements, helping to identify gaps, trends, and best practices. This foundational step is crucial for informing and guiding the development of new projects, ensuring they build upon and contribute to the existing body of knowledge.

2.1 Title: "Digitization of Printed ECG Data Using Image Processing Techniques"

Authors: Tahmida Tabassum and Mohiuddin Ahmad

Publication: Tabassum, Tahmida, and Mohiuddin Ahmad. "Numerical data extraction from ECG paper recording using image processing technique." 2020 11th international conference on electrical and computer engineering (ICECE). IEEE, 2020.

Abstract: Tahmida Tabassum and Mohiuddin Ahmad proposed a method for converting printed ECG data into digital signals using image processing techniques. The system digitizes scanned ECG strips from the MIT-BIH database and real patient records, enabling accurate reproduction of ECG images with reduced data size. This method facilitates high accuracy in digitization, improves the diagnosis of cardiac conditions, and supports further processing for automatic analysis and early detection of heart abnormalities. The research emphasizes the significance of digitizing ECG records for better storage, analysis, and clinical applications.

Methodology: The system employs advanced image processing techniques to convert scanned ECG strips from printed records into digital signals, allowing for efficient and accurate analysis. By leveraging datasets such as the MIT-BIH Arrhythmia Database, as well as real patient records, the system processes and digitizes ECG waveforms, preserving the integrity of the original data. These techniques involve steps like image segmentation, noise reduction, and feature extraction, which enhance the clarity and quality of the ECG signal for further processing. The digital conversion not only ensures high accuracy in reproducing the ECG images but also reduces the data size, making it more manageable for storage and efficient for real-time analysis.[1]

2.2 Title: "MIPAV (Medical Image Processing, Analysis, and Visualization)"

Authors: M.J. McAuliffe et al.

Publication: McAuliffe, Matthew J., et al. "Medical image processing, analysis and visualization in clinical research." Proceedings 14th IEEE symposium on computer-based medical systems. CBMS 2001. IEEE, 2001.

Abstract: The MIPAV (Medical Image Processing, Analysis, and Visualization) program is a platform-independent, general-purpose tool designed for clinical and quantitative analysis of medical images. This system facilitates 3D visualization and quantitative analysis of diverse image types, such as confocal microscopy, MRI, CT, and PET scans, on standard desktop computers. It eliminates the need for expensive UNIX workstations and supports remote collaboration and enhanced data sharing. By offering an affordable solution, MIPAV aids in studying, diagnosing, monitoring, and treating medical disorders.

Methodology: MIPAV is built to support 3D visualization and analysis of medical images on standard desktop systems. The program processes diverse medical image types using robust algorithms, ensuring precision and adaptability across various clinical and research applications. Additionally, the tool promotes remote collaboration and data sharing to enhance accessibility and affordability in medical image analysis. [2].

2.3 Title: "Review of Ensemble Learning Techniques for Disease Prediction"

Authors: Palak Mahajan et al.

Publication: Mahajan, Palak, et al. "Ensemble learning for disease prediction: A review." Healthcare. Vol. 11. No. 12. MDPI, 2023.

Abstract: Palak Mahajan et al reviewed ensemble learning techniques for disease prediction, focusing on four methods: bagging, boosting, stacking, and voting. The study analyzed 45 articles (2016–2023) that applied these techniques to five diseases: diabetes, skin disease, kidney disease, liver disease, and heart conditions. Stacking achieved the highest accuracy, particularly for skin disease and diabetes, followed by voting. Bagging performed best for kidney disease, while boosting excelled in liver disease and diabetes predictions. The research highlights the strengths of ensemble methods, including improved prediction accuracy, variability in performance across datasets, and valuable insights into selecting suitable ensemble models for disease analytics.

Methodology: The review focuses on analyzing ensemble learning methods—bagging, boosting, stacking, and voting—applied to disease prediction across multiple datasets. By examining 45 articles, the study identifies the performance of these techniques for various diseases and assesses their effectiveness. Stacking demonstrated superior accuracy in predicting skin disease and diabetes, while bagging and boosting excelled in kidney and liver disease predictions, respectively. [3].

2.4 Title: "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants"

Authors: Bauer and Kohavi

Publication: Bauer, Eric, and Ron Kohavi. "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Their Variants." Machine Learning, vol. 36, no. 1-2, 1999, pp. 105–139. Springer, doi:10.1023/A:1007515423169

Abstract: Bauer and Kohavi (1999) conducted an empirical study comparing voting classification algorithms, including bagging, boosting, and their variants. The research evaluated the performance of these ensemble techniques across various datasets, analyzing their strengths and weaknesses. Bagging demonstrated robustness in reducing variance, while boosting effectively reduced bias and enhanced accuracy. The study provided valuable insights into the trade-offs of different ensemble methods and emphasized the effectiveness of these techniques in improving classification performance. This foundational work serves as a guide for selecting suitable algorithms for specific tasks.

Methodology: The study involved a comparative analysis of voting classification algorithms, focusing on bagging, boosting, and their variants. Performance was assessed across diverse datasets, examining the ability of bagging to reduce variance and boosting to mitigate bias while improving accuracy. The research offered a comprehensive evaluation of these techniques' strengths and trade- offs, establishing their importance in enhancing classification performance [4].

2.5 Title: "Effectiveness of Ensemble in Observational Healthcare Data"

Authors: Behzad Naderalvojoud and Tina Hernandez-Boussard

Publication: Naderalvojoud, Behzad, and Tina Hernandez-Boussard. "Improving machine learning with ensemble learning on observational healthcare data." AMIA Annual Symposium

Proceedings. Vol. 2023. 2024.

Abstract: The study by Behzad Naderalvojoud and Tina Hernandez-Boussard explored the effectiveness of ensemble learning in the context of observational healthcare data, particularly in scenarios where individual models may have limitations due to the complexity and heterogeneity of healthcare data.

The researchers proposed an ensemble model designed to predict patients at risk of prolonged postoperative opioid use by combining two machine learning models, each trained on different covariates, such as patient demographics, surgical history, and pain management strategies. This innovative approach demonstrated significant improvements in prediction accuracy, as evidenced by enhanced AUROC (Area Under the Receiver Operating Characteristic curve) and AUPRC (Area Under the Precision-Recall curve) metrics, which are crucial for assessing the model's ability to correctly classify patients while minimizing false positives and false negatives. The ensemble model not only increased precision but also provided a deeper understanding of the various risk factors involved, offering healthcare providers more reliable tools for identifying patients at high risk of opioid dependency.

Methodology: The research introduced an ensemble learning framework aimed at predicting prolonged postoperative opioid use, a critical challenge in healthcare. The ensemble approach combined two distinct machine learning models, each trained on different sets of covariates, to leverage the strengths of each model and improve prediction accuracy. By integrating these models, the framework was able to capture a broader range of features and patterns within the data, thus enhancing its ability to predict outcomes accurately.

The performance of the ensemble method was evaluated using key metrics such as the Area Under the Receiver Operating Characteristic curve (AUROC) and the Area Under the Precision-Recall curve (AUPRC). These metrics are essential for assessing the model's ability to correctly identify patients at risk of prolonged opioid use while minimizing false positives and false negatives. The ensemble approach demonstrated significant improvements in both AUROC and AUPRC, highlighting its effectiveness in refining prediction precision and increasing reliability in healthcare settings.

CHAPTER 3

PRELIMINARIES

The preliminary requirements for the Cardiovascular Disease Prediction project involve several key aspects, focusing on the accurate prediction of cardiovascular diseases using ECG image processing and ensemble machine learning techniques. The project's scope includes developing a system that processes ECG images, extracts key features, applies dimensionality reduction, and predicts the likelihood of cardiovascular diseases using ensemble methods.

The implementation phase of the project involves several key steps:

1. **Preprocess ECG Images**: Convert ECG images into usable data.
2. **Extract Relevant Features**: Identify key ECG features such as wave amplitudes, intervals, and rhythms.
3. **Apply Dimensionality Reduction**: Reduce the dimensionality of the data to optimize the performance of machine learning models.
4. **Predict Disease Risk:** Use ensemble techniques such as Random Forest, Gradient Boosting, and XGBoost to predict cardiovascular disease risk.
5. **Provide Accurate Predictions**: Offer reliable predictions that can assist in early diagnosis and medical intervention.

3.1 Ensemble methods:

Ensemble methods are a class of machine learning algorithms that combine multiple models to improve the prediction accuracy and robustness. In this project, ensemble techniques are used to combine different machine learning models to predict cardiovascular disease risk.

Key Features:

- **Random Forest:** An ensemble method that constructs a multitude of decision trees and aggregates their results to improve prediction accuracy. It is less prone to overfitting and robust to noise.
- **Gradient Boosting**: An iterative technique that builds models sequentially, each new model correcting errors made by previous ones. It’s known for producing high- performance results.

- **XGBoost**: An optimized implementation of gradient boosting that is highly efficient and scalable, often producing state-of-the-art results in machine learning competitions.

3.2 ECG Image Processing:

It is central to this project as ECG data is often available in image format, either scanned or captured from digital systems. The images need to be preprocessed and converted into usable data for machine learning models.

Key Features:

- **Image Preprocessing**: This step includes converting color ECG images into grayscale and enhancing the image quality by removing noise and standardizing the image size.
- **Feature Extraction**: Using techniques like edge detection and thresholding to extract important ECG wave features, such as the P-wave, QRS complex, and T- wave.
- **Contour Detection:** Detecting the edges of the ECG waveform using methods like Canny edge detection, helps in extracting precise information about the shape and characteristics of the signal.

3.3 Dimensionality Reduction (PCA and t-SNE):

Dimensionality Reduction is a critical step to improve the computational efficiency and prediction performance of machine learning models.

Key Features:

- **PCA**: It reduces the number of features in the dataset while retaining most of the important variance, ensuring that essential information is preserved. By eliminating redundant or less significant features, it simplifies the dataset and improves computational efficiency. This reduction not only speeds up model training and inference but also enhances interpretability by focusing on the most influential features.
- **t-SNE:** Primarily used for data visualization in high-dimensional spaces, it helps in understanding the underlying structure of the data and improving model interpretability. t-Distributed Stochastic Neighbor Embedding (t-SNE) is primarily used for visualizing high-dimensional data in a lower-dimensional space, typically 2D or 3D. Its main strength lies in its ability to preserve the local structure of the data, making it particularly effective for identifying clusters.

CHAPTER 4

REQUIREMENT SPECIFICATIONS

This section documents the functional and non-functional requirements for the Cardiovascular Disease Prediction System using Ensemble Techniques.

4.1 Functional Requirements

The functional requirements for the Cardiovascular Disease Prediction System are:

1. Data Input and Preprocessing: The system must accept ECG image data or numeric features as input. The ECG images must be pre-processed to remove noise and enhance the clarity of the ECG waveform.
2. Feature Extraction:The system must extract key features from ECG images, including heart rate, P-wave, QRS complex, T-wave, and other clinically relevant intervals.
3. Ensemble Model Training: The system should train multiple machine learning models using ensemble methods such as Random Forest, Gradient Boosting, and XGBoost to predict the likelihood of cardiovascular diseases based on the extracted features.
4. Real-Time Prediction:The system must provide real-time predictions based on ECG data input. The predictions should classify the cardiovascular risk as low, medium, or high.
5. Model Evaluation:The system should evaluate the performance of the ensemble models using metrics like accuracy, precision, recall, F1-score, and ROC-AUC.
6. Data Visualization:The system should provide visualization of ECG features and prediction results. This includes displaying the processed ECG image and presenting classification results through visual tools like confusion matrices and ROC curves.
7. User Interface (UI):The system must provide an intuitive user interface that allows users to upload ECG data, view predictions, and interpret results easily.
   The UI should allow for the display of predicted risk levels and corresponding suggestions or warnings for further medical examination.
8. Report Generation:The system must generate detailed reports with prediction results, including ECG waveform analysis and risk classification.

By incorporating these functional requirements, the Cardiovascular Disease Prediction System will be capable of providing valuable insights into patient cardiovascular health, aiding in the early detection and monitoring of cardiovascular diseases.

4.2 Non-Functional Requirements

The non-functional requirements for the Cardiovascular Disease Prediction System are designed to ensure that the system delivers a high-quality, efficient, and sustainable user experience while maintaining its effectiveness over time. These requirements focus on various aspects such as usability, reliability, performance, design constraints, portability, and maintainability.

Usability is a critical factor in ensuring that both healthcare professionals and patients can use the system effectively. Healthcare professionals will benefit from a feature-rich interface that allows for detailed data analysis, comprehensive reporting, and customization of predictive models, enabling them to make informed decisions based on accurate and timely insights. Meanwhile, the patient interface will be simplified, focusing on ease of use, clear predictions, and accessible information, ensuring that users with limited technical expertise can interact with the system without difficulty. Furthermore, the system must be designed for ease of use, requiring minimal training, and should include help sections or tutorials to guide users in understanding the prediction results. This will help build user confidence and ensure that the system is user-friendly across all experience levels.

Reliability is fundamental to the operation of the system. The system must be highly reliable, with a focus on ensuring continuous availability and minimal downtime. It should be robust enough to handle large amounts of data without performance degradation or failures, even in environments with varying network speeds or hardware capabilities. This includes handling the stress of multiple users accessing the system simultaneously, especially in high- demand clinical settings, without compromising performance or accuracy. Reliability also extends to the system's accuracy in predicting cardiovascular disease risk, ensuring that predictions are consistently correct and dependable.

Regarding performance, the system must be optimized for real-time processing, with the ability to handle large datasets of ECG data efficiently. The prediction engine should deliver results within seconds of receiving the input data, allowing for quick decision-making in clinical contexts. The response time should be between 3 to 5 seconds from data input to prediction output, ensuring that the system remains responsive even with large volumes of input data. This is crucial for scenarios where rapid decision-making can have a significant impact on patient outcomes, such as emergency or high-pressure clinical environments.

Design constraints further guide the development of the system, particularly with regard to its user base and technology stack. The system must be able to serve both healthcare professionals, who need advanced features for diagnosis and monitoring, and patients, who require simple and clear results. This dual-user approach means the system's interface and features must be carefully considered to meet the needs of each group without causing confusion or unnecessary complexity. Technologically, the development will leverage Python, along with well-established machine learning frameworks like Scikit-learn, TensorFlow, and Keras, to build and train predictive models that can handle complex ECG data. These tools are ideal for creating scalable, high-performance models and algorithms capable of processing real-world clinical data.

4.3 Hardware Requirements

The hardware requirements of the Project are:

1. Processor: Minimum Intel Core i5 or equivalent processor for smooth performance during model training and prediction.
2. Memory (RAM): Minimum 8 GB of RAM to handle large datasets and ensure smooth performance during analysis.
3. Graphics Processing Unit (GPU): A dedicated GPU (e.g., NVIDIA GTX 1060 or higher) is recommended for model training, especially for deep learning models.
4. Storage: Minimum 100 GB of free disk space for storing ECG datasets, machine learning models, and output results.
5. An internet connection is needed to upload datasets and share results with external tools.

4.4 Software Requirements

The Cardiovascular Disease Prediction System will be developed to be compatible with major operating systems, including Windows, Linux, and macOS, ensuring broad accessibility and flexibility for deployment in various environments. The core development will use Python (version 3.6 or higher), with key libraries such as Scikit-learn for traditional machine learning models, TensorFlow/Keras for deep learning applications, and OpenCV for image processing tasks related to ECG image analysis. For web development, the system will utilize a web framework like Django or Flask for the backend, providing robust, scalable, and secure server- side functionality. The frontend will be built using React or Angular, offering a dynamic and responsive user interface for both healthcare professionals and patients. In terms of machine learning, the system will

integrate libraries like XGBoost for ensemble learning models, Scikit- learn for a wide range of traditional machine learning algorithms, and TensorFlow/Keras for deep learning models, enabling the system to effectively predict cardiovascular disease risk from ECG data using the most advanced algorithms available. These technologies together will ensure the system is efficient, scalable, and capable of delivering high-performance predictions and real-time results.

4.5 User Requirements

4.5.1 User Profiles

Users should be able to create and manage profiles to store personal health data and prediction history, without relying on a database. Local storage on the system can be used for saving the profiles. The profiles should store customization options, such as preferred output formats and frequency of prediction notifications.

4.5.2 Help and Support

The system must provide a help section with detailed documentation on how to use the system, interpret results, and troubleshoot issues. Users should have access to a support team for any questions or technical issues. Additionally, the help section should include FAQs, step-by-step guides, and video tutorials to ensure ease of use for all types of users.

4.5.3 Feedback Mechanism

The system should incorporate a robust feedback mechanism to ensure continuous improvement in both accuracy and user satisfaction. Users, including healthcare professionals and patients, should be able to provide feedback on the accuracy and relevance of the predictions. This mechanism can include options for users to report incorrect predictions, suggest areas for improvement, or highlight any issues with the user interface or functionality. All feedback should be logged and categorized, enabling the development team to prioritize and address critical issues. Periodic reviews of the collected feedback will help identify recurring patterns or areas requiring enhancement, such as refining predictive models, improving the user interface, or adding new features. By integrating user insights into the system's development lifecycle, this mechanism will support ongoing optimization of performance, usability, and overall reliability.

CHAPTER 5

SYSTEM DESIGN

System design involves explaining the architectures, component, module, interface and data for a method that satisfies the specific requirement and modelling is a method of creating a simplied representation of a system to understand and communicate its structure and behaviour.

5.1 High Level Design

The high-level design includes:

5.1.1 Data Collection & Preprocessing

The steps of Data Collection & Preprocessing are:

- Gather patient data including ECG images, demographic data (e.g., age, gender), and lifestyle factors (e.g., smoking, diet, exercise).
- Input ECG Data: If ECG data is in the form of images, the images will be captured and processed using image processing techniques (e.g., grayscale conversion, resizing, edge detection).
- If ECG data is numerical, it will be directly used for further preprocessing steps.
- Preprocessing: Normalize ECG data and extract key features (e.g., heart rate, PQRST complex).
- Feature Extraction: Key features from the ECG (or processed images) will be extracted using methods like HOG, FFT, or statistical features.

5.1.2 Ensemble Model (Training and Prediction)

- Prepare the preprocessed data (feature vectors) for training the ensemble models.
- Model Selection: The system will use an ensemble approach with multiple machine learning models like SVM, Random Forest, Logistic Regression, and KNN.
- Cross-validation will be used to avoid overfitting and ensure reliable results.

Ensemble Techniques:

- Bagging: Train each model on a different subset of the data to reduce variance.
- Boosting: Focus on misclassified instances to improve prediction accuracy.

5.1.3 Post-Processing and Result Interpretation

The Post-Processing and Result Interpretation includes:

- Take the model output (predicted risk).
- Risk Interpretation: Convert the prediction into a human-readable format such as “Low Risk”, “Medium Risk”, or “High Risk”.
- If High Risk, the system will display additional medical recommendations or actions (e.g., consult a cardiologist).
- Output: The user will receive a clear, easy-to-understand output regarding their cardiovascular risk, as well as appropriate suggestions for their health management.

5.1.4 User Interface (UI)

User provides data (e.g., upload ECG image, fill in personal details like age and lifestyle).

- Input Data: The system allows the user to upload ECG images or enter numerical ECG data.
- The user provides additional information such as age, gender, and lifestyle factors to improve prediction accuracy.
- Output Data: After processing, the system displays the risk prediction and interpretation (Low, Medium, High Risk), along with personalized recommendations.
- User Interaction: Users can click on buttons for new prediction, exit, or get more details about the prediction.

5.1.5 Model Evaluation and Feedback:

After a prediction is made, the system evaluates the performance of the ensemble model based on actual outcomes.

- Model Performance: Regular performance evaluation using metrics such as accuracy, precision, recall, and F1-score.
- The system will compare the prediction results with the actual diagnosis (if available) to check the accuracy.
- Feedback: With feedback of the users, the model will be further improvised.

5.2 System Architecture

The Cardiovascular Disease Prediction System is designed using a two-tier architecture, which consists of two main components: the presentation tier and the application tier. In this architecture, the presentation tier is responsible for interacting with the user. It allows users to input data, such as ECG images or health information, and displays the prediction results, which indicate the likelihood of cardiovascular disease. The user interface is accessible via a desktop or web application, making it easy for users to provide necessary input and receive output.

The application tier handles the core processing and prediction logic. This tier is responsible for preprocessing the data, which includes tasks like image resizing, grayscale conversion, and feature extraction. Once the data is processed, machine learning models, such as ensemble techniques (e.g., SVM, Random Forest, and KNN), are used to predict the cardiovascular risk based on the input features. The results are then sent back to the presentation tier for display.

This two-tier architecture is efficient for real-time processing as it directly connects the user interface with the server. The absence of a database simplifies the design, reducing both infrastructure requirements and complexity. This architecture is well-suited for small to medium-scale applications where real-time prediction and ease of use are critical. The two-tier architecture also provides flexibility for future enhancements. As the system evolves, additional components such as a database tier can be integrated to store historical data, user logs, or model performance metrics. This would enable advanced functionalities like personalized predictions, longitudinal tracking of patient health, and continuous improvement of the prediction models through retraining on new data. Furthermore, the architecture can be scaled to include cloud-based deployment.

Additionally, the presentation tier and application tier work together to ensure efficient processing and seamless interaction. The presentation tier provides a simple interface for inputting data, such as ECG images and health details, while displaying prediction results. The application tier handles core processing, including image preprocessing and feature extraction, before applying ensemble- based machine learning models like SVM, Random Forest, and KNN for accurate predictions. This two-tier architecture ensures real-time disease prediction while keeping the system lightweight and adaptable for future enhancements without major modifications.

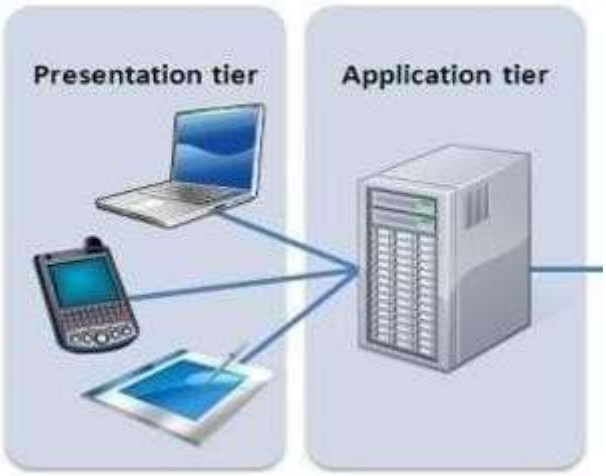

Figure 5.2.1: Two-tier architecture

The user interface is implemented on any platform such as a desktop PC, smartphone, or tablet as a native application, web app, mobile app, voice interface, etc. It uses a standard graphical user interface with different modules running on the application server. The user uses the front-end to input images to input gestures and speech for translation.

The relational database management system on the database server contains the computer data storage logic. The middle tiers are usually multitiered containing core logic for processing and logic which will be developed using Python and Django.

Since the three are not physical but logical in nature, they may run in different servers both in on- premises based solutions, as well as in Software-as-a-Service (SaaS).

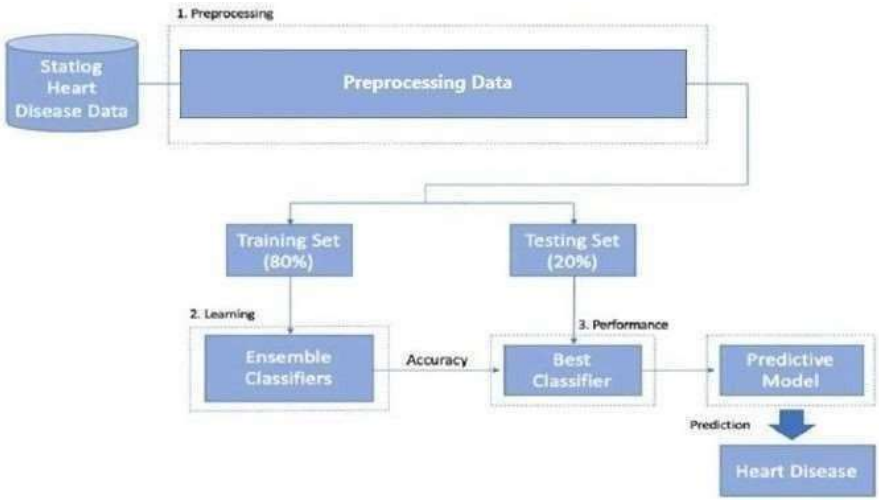

Figure 5.2.2: System Architecture of Cardiovascular Disease Prediction System

5.3 Use-case diagram

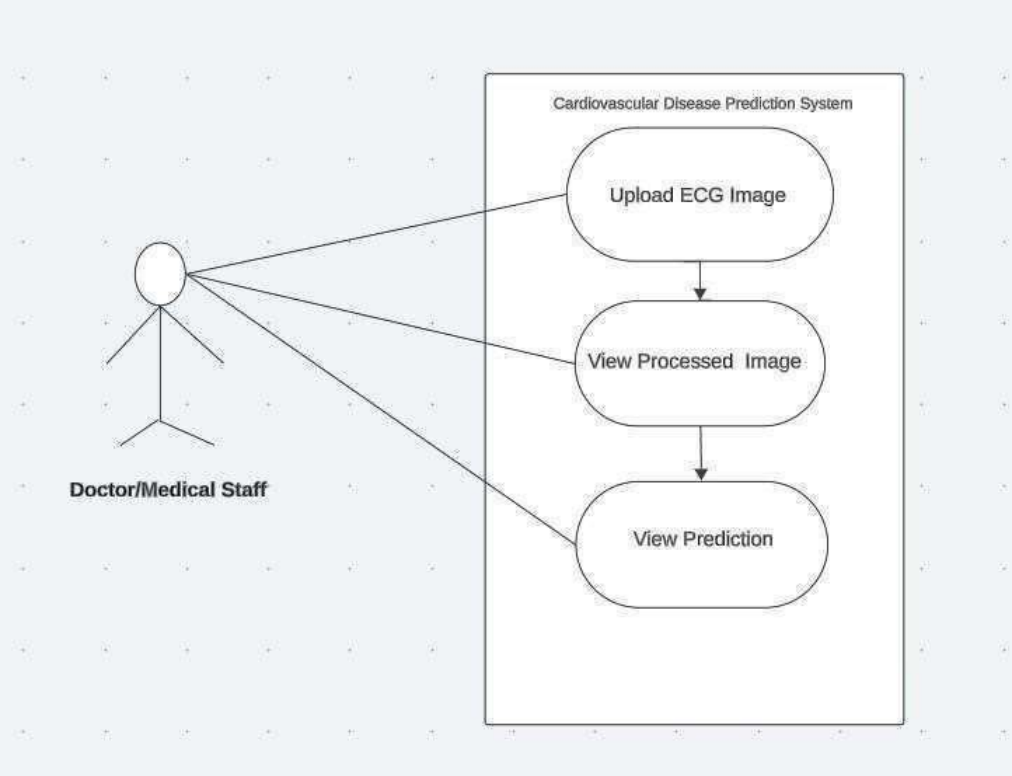

Figure 5.3: Use-Case Diagram of Cardiovascular Disease Prediction System

A **use case diagram** is a critical tool in system design, illustrating the interactions between different actors (users or systems) and the functionalities of the system being developed. For the Cardiovascular Disease Prediction System, the diagram would visually represent how healthcare professionals and patients interact with various features of the system. Key use cases for healthcare professionals might include uploading ECG data, accessing preprocessing tools to clean and enhance the data, viewing feature extraction results, running advanced predictive models, evaluating model performance through metrics like accuracy and ROC-AUC, visualizing results, and generating detailed reports for patient diagnosis and treatment planning. On the other hand, patients would interact with more user-friendly functionalities such as uploading ECG or health data, viewing simplified prediction results (e.g., low, medium, or high cardiovascular risk), and receiving health recommendations or alerts for follow-up actions.

The use case diagram also captures the system's backend processes, such as data preprocessing, feature extraction, and model training, which occur transparently to users but are essential for the system's operation.

CHAPTER 6

IMPLEMENTATION

6.1 Overview of Technologies Used

Technologies used are:

The technologies used in the project are:

1. **Python:** Python is a high-level, interpreted programming language that emphasizes readability and simplicity. It is widely used in data science and machine learning due to its rich ecosystem of libraries and tools. Python 3 is used in this project for data preprocessing, machine learning model development, and integrating various components of the system.

2. **Scikit-learn (Sklearn):** Scikit-learn is a powerful Python library for machine learning that provides simple and efficient tools for data mining and data analysis. It is used in this project for implementing machine learning models such as Decision Trees, Random Forests, and Support Vector Machines (SVM). It simplifies tasks like model training, testing, and evaluation, making it ideal for the ensemble approach used in this system.

3. **Pandas:** Pandas is a Python library used for data manipulation and analysis. It provides powerful data structures like DataFrames, which are ideal for handling structured data. In this project, Pandas is used to clean, preprocess, and analyze medical datasets, ensuring that the data is in the proper format for machine learning algorithms.

4. **NumPy:** NumPy is an essential Python library for numerical computations. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions. In this project, NumPy is used to perform mathematical operations on the data, ensuring efficient computations during the training and evaluation of machine learning models.

5. **Matplotlib and Seaborn:** These Python libraries are used for data visualization. Matplotlib allows for creating static, interactive, and animated visualizations, while Seaborn is built on top of Matplotlib and provides enhanced statistical visualizations.

6.2 Implementation

The modules of the project are:

Module 1: Frontend (Streamlit)

The frontend of the system is developed using Streamlit, a powerful Python library that enables the rapid creation of interactive web applications. The Streamlit module serves as the user interface for the entire project, enabling users to interact with the system and view the results. The key functionalities of the frontend module include:

1) **Image Upload:** Users can upload ECG images, which will then be processed and analyzed by the backend system.
2) **Prediction Display:** After the model processes the image, the frontend displays the prediction results for cardiovascular diseases.
3) **User Interaction:** The interface includes intuitive controls for users to interact with the system, view explanations of predictions, and download or view the results.
4) **Visualization:** Streamlit allows for easy visualization of ECG images, prediction results, and possibly even the intermediate processing steps.This module bridges the gap between the backend logic (image processing and prediction) and the end user, offering an easy-to-use interface to visualize predictions and interact with the system.

Module 2: Image Processing

The Image Processing Module is responsible for transforming the raw ECG images into a form that can be processed by machine learning algorithms. This step is crucial as ECG data can vary in format, quality, and resolution, and needs to be standardized for effective analysis. The main operations in this module include:

1. **Preprocessing:** ECG images are converted to grayscale, resized to a standard dimension, and enhanced to remove noise. This step ensures that the images are in an optimal format for feature extraction.
2. **Feature Extraction**: Techniques such as edge detection, filtering, and segmentation are applied to extract key features from the ECG images. These features represent important aspects of heart activity that are crucial for disease prediction.
3. **Image Normalization**: The processed image is normalized to ensure consistent lighting and contrast, helping the model focus on the essential patterns rather than being distracted

by irrelevant noise.

4. **Image to Feature Vector Conversion:** After processing, the images are transformed into feature vectors that represent the relevant data from the images and can be fed into the prediction model.

This module acts as the preprocessing step to ensure that the data is ready for use in model prediction.

Module 3: Model Creation and Prediction (Ensemble Technique)

The Model Creation and Prediction Module focuses on using machine learning techniques to predict cardiovascular diseases based on the features extracted from the ECG images. The core of this module is the Ensemble Learning method, which combines the predictions of multiple models to improve accuracy and robustness. The steps involved include:

Model Training: Various machine learning classifiers, such as Decision Trees, Random Forest, and Support Vector Machines (SVM), are trained on the feature vectors extracted from ECG images. These models analyze the data to learn the distinguishing patterns associated with different cardiovascular conditions. During the training phase, the models adjust their internal parameters to optimize classification accuracy by recognizing key characteristics of both healthy and diseased heart conditions. This training process involves multiple iterations, refining the models to improve their predictive performance on unseen data.

1. **Ensemble Technique:** The module uses ensemble learning techniques such as Voting Classifier or Stacking to combine the predictions of the individual models. This helps in improving the overall performance by reducing errors from individual classifiers. By leveraging multiple models, the system ensures better generalization and minimizes the impact of outliers.
2. **Prediction:** Once the model is trained, it predicts whether the input ECG image corresponds to a healthy or diseased heart. The ensemble model considers all trained models' outputs and selects the final prediction based on majority voting or weighted voting.
3. **Result Output:** The system outputs the prediction (such as “Healthy” or “At risk of cardiovascular disease”) along with a confidence score that indicates the model’s certainty about the prediction.

These three modules work together to provide an integrated system for predicting cardiovascular diseases based on ECG images. The frontend (Streamlit) enables user interaction, while image processing ensures the data is ready for prediction. Finally, the ensemble model uses various algorithms to generate an accurate and robust prediction.

6.3 Difficulties encountered and Strategies used

One of the key challenges we faced during the Cardiovascular Disease Prediction project was gathering high-quality, labeled ECG data. Publicly available datasets were limited and often imbalanced, with far more records of healthy individuals than those with cardiovascular conditions. To overcome this, we leveraged existing datasets like the PhysioNet and MIT-BIH Arrhythmia Database and applied data augmentation techniques to increase the dataset size. Additionally, we employed preprocessing methods to reduce noise and standardize the ECG images, improving feature extraction. The class imbalance was addressed using resampling techniques and ensemble learning methods, which helped the model perform better despite the unequal distribution of data. Despite the difficulties, these strategies allowed us to create a reliable dataset for model training and improve prediction accuracy.

6.4 Pseudo-code for Pre-processing Techniques

6.4.1 Frontend Code

```
import streamlit as st from Ecg
import ECG st.set_page_config(
page_title="Cardiovascular Disease Prediction", page_icon=" ☑
    ",
layout="wide"
)
st.markdown("""
<style>
.stApp {
background: linear-gradient(135deg, #f6d365 0%, #fda085 100%); font-
    family: Arial, sans-serif;
}
```

```
.stExpander {
background-color: rgba(255, 255, 255, 0.8);
    border-radius: 10px;
margin-bottom: 10px;
box-shadow: 0 4px 6px rgba(0,0,0,0.1);
}
h1, h4, p, .stMarkdown, .caption-text {
    color: #333333;
}
.bold-text {
font-weight: bold; font-
    size: 1.2em; color:
    #333333;
}
</style>
""", unsafe_allow_html=True) st.markdown("""
<div style="background-color: rgba(255, 255, 255, 0.8); padding: 20px; border-radius: 10px;
    text-align: center;">
<h1>Cardiovascular Disease Prediction Using Ensemble Technique</h1>
</div>
""", unsafe_allow_html=True) ecg =
    ECG()
st.markdown("""
<div style="background-color: rgba(255, 255, 255, 0.8); padding: 20px; border-radius: 10px;
    margin-bottom: 20px; text-align: center;">
<h4>Upload your ECG image for analysis</h4>
</div>
""", unsafe_allow_html=True)
uploaded_file = st.file_uploader("Choose a file", type=['png', 'jpg', 'jpeg']) # if
    uploaded_file is not None:
#   try:
```

```
#           st.markdown("### Uploaded Image")
#           ecg_user_image_read = ecg.getImage(uploaded_file)
# st.image(ecg_user_image_read, caption="Uploaded ECG Image", use_column_width=True) #
              with st.expander("Gray Scale Image"):
#           ecg_user_gray_image_read = ecg.GrayImgae(ecg_user_image_read)
#           st.image(ecg_user_gray_image_read, caption="Grayscale ECG Image",
            use_column_width=True)
if uploaded_file is not None:
            try:
st.markdown("### Uploaded Image") ecg_user_image_read =
            ecg.getImage(uploaded_file)
st.image(ecg_user_image_read, caption="", use_column_width=True) # Apply
            the same caption style
st.markdown("<p style='color: black; font-size: 16px; font-weight: bold; text-align:
            center;'>Uploaded ECG Image</p>", unsafe_allow_html=True)

with st.expander("Gray Scale Image"):
ecg_user_gray_image_read = ecg.GrayImgae(ecg_user_image_read)
            st.image(ecg_user_gray_image_read, caption="", use_column_width=True) #
            Apply the same caption style for grayscale image
st.markdown("<p style='color: black; font-size: 16px; font-weight: bold; text-align:
            center;'>Grayscale ECG Image</p>", unsafe_allow_html=True)
with st.expander("Dividing Leads"):
dividing_leads = ecg.DividingLeads(ecg_user_image_read) col1, col2 =
            st.columns(2)
with col1:
st.image('Leads_1-12_figure.png', use_column_width=True)
st.markdown("""                    <p style='color: black; font-size: 16px; font-weight: bold; text-
            align: center;'>Long Lead 1-12</p> """, unsafe_allow_html=True)
with col2:
st.image('Long_Lead_13_figure.png',  use_column_width=True)
```

```
st.markdown("""                    <p style='color: black; font-size: 16px; font-weight: bold; text-
        align: center;'>Long Lead 13</p> """, unsafe_allow_html=True)
# Use st.markdown to display the caption with HTML and custom styling
        with st.expander("Preprocessed Leads"):
ecg_preprocessed_leads = ecg.PreprocessingLeads(dividing_leads) col1, col2
        = st.columns(2)
with col1:
st.image('Preprossed_Leads_1-12_figure.png', caption="", use_column_width=True) # Apply
        caption style for Preprocessed Leads 1-12
st.markdown("<p style='color: black; font-size: 16px; font-weight: bold; text-align:
        center;'>Preprocessed Leads 1-12</p>", unsafe_allow_html=True)
with col2:
st.image('Preprossed_Leads_13_figure.png', caption="", use_column_width=True) # Apply
        caption style for Preprocessed Lead 13
st.markdown("<p style='color: black; font-size: 16px; font-weight: bold; text-align:
        center;'>Preprocessed Lead 13</p>", unsafe_allow_html=True)
with st.expander("Extracting Signals"):
ec_signal_extraction = ecg.SignalExtraction_Scaling(dividing_leads)
        st.image('Contour_Leads_1-12_figure.png', caption="", use_column_width=True)
# Apply caption style for Signal Contours
st.markdown("<p style='color: black; font-size: 16px; font-weight: bold; text-align:
        center;'>Signal Contours</p>", unsafe_allow_html=True)
# with st.expander("Preprocessed Leads"):
# ecg_preprocessed_leads = ecg.PreprocessingLeads(dividing_leads)

#  col1, col2 = st.columns(2)
#  with col1:
#  st.image('Preprossed_Leads_1-12_figure.png', caption="<p class='caption-
            text'>Preprocessed Leads 1-12</p>", use_column_width=True)
#   with col2:
#  st.image('Preprossed_Leads_13_figure.png', caption="<p class='caption- text'>Preprocessed
            Lead 13</p>", use_column_width=True)
```

```
# with st.expander("Extracting Signals"):
# ec_signal_extraction = ecg.SignalExtraction_Scaling(dividing_leads)
# st.image('Contour_Leads_1-12_figure.png', caption="<p class='caption-text'>Signal
        Contours</p>", use_column_width=True)
with st.expander("1D Signal Conversion"): ecg_1dsignal =
        ecg.CombineConvert1Dsignal() st.write(ecg_1dsignal)
with st.expander("Dimensionality Reduction"): ecg_final =
        ecg.DimensionalReduciton(ecg_1dsignal)
        st.write(ecg_final)
with st.expander("Prediction Results"): ecg_model =
        ecg.ModelLoad_predict(ecg_final)
st.markdown(f"<p class='bold-text'>{ecg_model}</p>", unsafe_allow_html=True) except
        Exception as e:
st.error(f"An error occurred during processing: {e}")
        st.markdown("""
<div style="text-align: center; margin-top: 50px;">
<p>© 2024 Cardiovascular Disease Prediction Using Ensemble Technique</p>
</div>
""", unsafe_allow_html=True)
```

6.4.2 Image Processing and Model Building

```
from skimage.io import imread
from skimage import color
import matplotlib.pyplot as plt
from skimage.filters import threshold_otsu,gaussian
from skimage.transform import resize
from numpy import asarray
from skimage.metrics import structural_similarity
from skimage import measure
from sklearn.decomposition import PCA
from sklearn.neighbors import KNeighborsClassifier
```

```
import joblib
from sklearn.preprocessing import MinMaxScaler
import pandas as pd
import numpy as np
import os
from natsort import natsorted
from sklearn import linear_model, tree, ensemble
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
class ECG:
    def getImage(self,image):
        """
        this functions gets user image
        return: user image
        """
        image=imread(image)
        return image
    def GrayImgae(self,image):
        """
        This funciton converts the user image to Gray Scale
        return: Gray scale Image
        """
        image_gray = color.rgb2gray(image)
        image_gray=resize(image_gray,(1572,2213))
        return image_gray
    def DividingLeads(self,image):
        """
```

- This Funciton Divides the Ecg image into 13 Leads including long lead. Bipolar limb
- Leads(Leads1,2,3). Augmented unipolar limb leads(aVR,aVF,aVL). Unipolar (+) chest leads(V1,V2,V3,V4,V5,V6)

```
        return : List containing all 13 leads divided
        """
```

```
        Lead_1 = image[300:600, 150:643] # Lead 1
        Lead_2 = image[300:600, 646:1135] # Lead aVR
        Lead_3 = image[300:600, 1140:1625] # Lead V1
        Lead_4 = image[300:600, 1630:2125] # Lead V4
        Lead_5 = image[600:900, 150:643] #Lead 2
        Lead_6 = image[600:900, 646:1135] # Lead aVL
        Lead_7 = image[600:900, 1140:1625] # Lead V2
        Lead_8 = image[600:900, 1630:2125] #Lead V5
        Lead_9 = image[900:1200, 150:643] # Lead 3
        Lead_10 = image[900:1200, 646:1135] # Lead aVF
        Lead_11 = image[900:1200, 1140:1625] # Lead V3
        Lead_12 = image[900:1200, 1630:2125] # Lead V6
        Lead_13 = image[1250:1480, 150:2125] # Long Lead
        #All Leads in a list
Leads=[Lead_1,Lead_2,Lead_3,Lead_4,Lead_5,Lead_6,Lead_7,Lead_8,Lead_9,Lead_1
0,Lead_11,Lead_12,Lead_13]
        fig , ax = plt.subplots(4,3)
        fig.set_size_inches(10,  10)
        x_counter=0
        y_counter=0
        #Create 12 Lead plot using Matplotlib subplot
        for x,y in enumerate(Leads[:len(Leads)-1]):
            if (x+1)%3==0:
                ax[x_counter][y_counter].imshow(y)
                ax[x_counter][y_counter].axis('off')
                ax[x_counter][y_counter].set_title("Leads {}".format(x+1))
                x_counter+=1
                y_counter=0
            else:
                ax[x_counter][y_counter].imshow(y)
                ax[x_counter][y_counter].axis('off')
                ax[x_counter][y_counter].set_title("Leads {}".format(x+1))
```

```
                        y_counter+=1
            #save the image
            fig.savefig('Leads_1-12_figure.png')
            fig1 , ax1 = plt.subplots()
            fig1.set_size_inches(10, 10)
            ax1.imshow(Lead_13)
            ax1.set_title("Leads 13")
            ax1.axis('off')
            fig1.savefig('Long_Lead_13_figure.png')
            return Leads
    def PreprocessingLeads(self,Leads):
            """
            This Function Performs preprocessing to on the extracted leads.
            """
            fig2 , ax2 = plt.subplots(4,3)
            fig2.set_size_inches(10,  10)
            #setting counter for plotting based on value
            x_counter=0
            y_counter=0
            for x,y in enumerate(Leads[:len(Leads)-1]):
                    #converting to gray scale
                    grayscale = color.rgb2gray(y)
                    #smoothing image
                    blurred_image = gaussian(grayscale, sigma=1)
                    #thresholding to distinguish foreground and background
                    #using otsu thresholding for getting threshold value
                    global_thresh = threshold_otsu(blurred_image)
                    #creating binary image based on threshold
                    binary_global = blurred_image < global_thresh
                    #resize image
                    binary_global = resize(binary_global, (300, 450))
.savefig('Preprossed_Leads_1-12_figure.png')
#plotting lead 13
```

```
fig3 , ax3 = plt.subplots() fig3.set_size_inches(10,
10) #converting to gray scale
grayscale = color.rgb2gray(Leads[-1]) #smoothing
image
blurred_image = gaussian(grayscale, sigma=1) #thresholding to
distinguish foreground and background #using otsu thresholding for
getting threshold value global_thresh = threshold_otsu(blurred_image)
print(global_thresh)
#creating binary image based on threshold binary_global =
blurred_image < global_thresh
ax3.imshow(binary_global,cmap='gray') ax3.set_title("Leads
13")
ax3.axis('off') fig3.savefig('Preprossed_Leads_13_figure.png')

    def  SignalExtraction_Scaling(self,Leads):
        """
        This Function Performs Signal Extraction using various steps,techniques: conver to grayscale, apply gaussian filter, thresholding, perform contouring to extract signal image and then save the image as 1D signal
        """
        fig4 , ax4 = plt.subplots(4,3)
        #fig4.set_size_inches(10,  10)
        x_counter=0
        y_counter=0
        for x,y in enumerate(Leads[:len(Leads)-1]):
            #converting to gray scale
            grayscale = color.rgb2gray(y)
            #smoothing image
            blurred_image = gaussian(grayscale, sigma=0.7)
            #thresholding to distinguish foreground and background
            global_thresh = threshold_otsu(blurred_image)
            #creating binary image based on threshold
            binary_global = blurred_image < global_thresh
```

```
fig.savefig('Contour_Leads_1-12_figure.png') def CombineConvert1Dsignal(self):
 """
 This function combines all 1D signals of 12 Leads into one File, csv for model returns the final dataframe
 """

 #first read the Lead1 1D signal
 test_final=pd.read_csv('Scaled_1DLead_1.csv') location= os.getcwd()
 print(location)

 for files in natsorted(os.listdir(location)): if
        files.endswith(".csv"):
              if files!='Scaled_1DLead_1.csv': df=pd.read_csv('{}'.format(files))


test_final=pd.concat([test_final,df],axis=1,ignore_index=True) _final


def DimensionalReduciton(self,test_final): """
This function reduces the dimensinality of the 1D signal using PCA returns the final
 dataframe
"""
pca_loaded_model = joblib.load('PCA_ECG (1).pkl') result =
 pca_loaded_model.transform(test_final) final_df =
 pd.DataFrame(result)
return final_df

def ModelLoad_predict(self,final_df): """
This Function Loads the pretrained model and perfrom ECG classification return the
        classification Type. """

loaded_model = joblib.load('Heart_Disease_Prediction_using_ECG (4).pkl') result =
        loaded_model.predict(final_df)
if result[0] == 1:
return "You ECG corresponds to Myocardial Infarction" elif result[0] == 0:
return "You ECG corresponds to Abnormal Heartbeat" elif result[0] == 2:
return "Your ECG is Normal"
else:
return "You ECG corresponds to History of Myocardial Infarction"
```

CHAPTER 7

TESTING AND EVALUATION

Testing and evaluation are crucial steps in ensuring the reliability and accuracy of the Cardiovascular Disease Prediction system. These processes involve validating the system's functionality, performance, and usability to meet the desired requirements.

7.1 Unit Testing

Unit testing is the process of verifying individual components or modules of a software application to ensure they function as expected. Each unit test focuses on a specific functionality of a single module, isolating it from other parts of the system. The purpose is to catch and fix bugs early in the development process, improving the reliability and quality of the software.

7.1.1. Objectives of Unit Testing

The primary objective of unit testing is to ensure the correctness, reliability, and robustness of individual components or modules in a software application. It focuses on validating the smallest functional parts of the system in isolation to detect and fix bugs early in the development cycle. Specifically, the objectives are:

1. **Verification of Module Functionality:**

   To confirm that each module behaves as expected and produces correct outputs for given inputs.

2. **Early Bug Detection:**

   To identify and fix errors or issues in the code at an early stage, reducing the cost and complexity of debugging in later phases.

3. **Ensuring Code Quality:**

   To improve code quality by ensuring each function, method, or component meets predefined specifications and handles various scenarios, including edge cases.

4. **Facilitating Code Refactoring:**

   To provide a safety net for developers when modifying or optimizing code, ensuring that changes do not break existing functionality.

5. **Supporting Integration:**
   To prepare individual modules for seamless integration into the larger system verifying their correctness and compatibility.
6. **Improving Reliability:**
   To increase the overall reliability of the software by addressing issues at the module level before they propagate to higher levels.

Table 7.1: Unit Test Cases

TEST CASE	MODULE	INPUT DESCRIPTION	EXPECTED RESULT	OBTAINED RESULT	STATUS
1.	FRONT-END (Streamlit)	Upload valid ECG image	ECG image displays correctly	ECG image displayed correctly	PASS
2.	Model Prediction	Evaluate the ensemble model on test data	Ensemble model generated	Training metrics working	PASS
3.	Image Processing	Provide an ECG image	Image processed and grayscale image returned	Grayscale image displayed	PASS
4.	Image Processing	Upload high-resolution ECG image	Contours detected and highlighted	Contours detected	PASS
5.	Image Processing	Apply binarization to ECG image	Binary image is generated	Binary image generated	PASS

an overview of unit test cases conducted to validate various components of a system involving a front-end interface, model prediction, and image processing functionalities. Each test case specifies the module under evaluation, the input provided, the expected outcome, the obtained result, and the test status. For the front-end module, the test ensured that a valid ECG image uploaded by the user displayed correctly, which it successfully achieved.

7.1.2. Detailed Explanation of Test Case 1

The **front-end image upload and display function** is a critical component of the Cardiovascular Disease Prediction system, ensuring the uploaded ECG images are validated, processed, and displayed correctly before further analysis.This test case aims to evaluate the system's ability to handle image uploads efficiently, ensuring error-free operations for both valid and invalid inputs. The system must demonstrate robustness, user-friendliness, and precision in handling image data.

This test case involves the following steps:

Objective:

Verify that the front-end correctly processes and displays a valid ECG image by validating the file format, rendering it in the application interface, and ensuring its clarity.

Verify that:

The image is validated for supported formats (.png, .jpg) and file integrity. The uploaded ECG image is rendered properly in the Streamlit interface, matching the original dimensions and maintaining clarity. Errors are correctly handled for invalid or unsupported file uploads, with appropriate error messages displayed. The rendered image matches the uploaded image in quality and dimensions.

Steps:

Upload a valid ECG image in supported formats through the Streamlit interface. Confirm that the system validates the file format and prevents corrupted or unsupported files. Display the uploaded image in the front-end interface without cropping or distortion. Compare the displayed image to the original file and ensure consistency.

This test case ensures that the front-end of the Cardiovascular Disease Prediction system is both robust and user-friendly when handling ECG image uploads. It is critical for the system to allow users to seamlessly upload valid ECG images in formats like .png and .jpg while preventing any unsupported or corrupted files from being uploaded. Once an image is uploaded, the system will confirm that the file format is compatible and that the image is not corrupted. The image should be displayed exactly as it is without any distortion.

7.2 Integration Testing

Integration testing ensures the seamless interaction between the front-end, image processing, and model prediction modules of our Cardiovascular Disease Prediction System.

a. Front-End and Image Processing Integration

Verify that when an ECG image is uploaded through the Streamlit front-end, it is successfully passed to the image processing module for preprocessing (e.g., resizing, grayscale conversion).

b. Image Processing and Model Prediction Integration

The integration of image processing and model prediction is a crucial component of the Cardiovascular Disease Prediction system. This process involves seamlessly linking the steps of uploading, validating, and displaying an ECG image to the model's predictive capabilities. After the image is successfully uploaded and rendered in the system, it undergoes a series of preprocessing steps, such as normalization, noise reduction, and feature extraction.

7.3 System Testing

System testing evaluates the complete, integrated Cardiovascular Disease Prediction System to ensure it meets specified requirements. It involves testing the system’s functionality, performance, security, and compatibility.

7.4 User Acceptance Testing

User Acceptance Testing (UAT) for our Cardiovascular Disease Prediction System ensures that the system meets end-user expectations and functional requirements by focusing on key usability and performance aspects. The UAT process involves validating the front-end interface to confirm it is intuitive, user-friendly, and accessible, ensuring users can easily upload ECG images through Streamlit without any complications. It also includes verifying that the system’s image processing functionality operates smoothly by accurately converting ECG images to grayscale and displaying the processed images correctly. Additionally, the end-to-end process, from image upload to prediction output, should function seamlessly, delivering accurate and timely results while maintaining an optimal user experience.

CHAPTER 8

RESULT

8.1 Experimental Dataset

In the Experimental Dataset section, we have sourced the ECG data from the Mendeley Data repository (ECG Images dataset of Cardiac Patients - Mendeley Data), which is a widely recognized platform for sharing research datasets. This repository offers a diverse collection of ECG data, including records from both healthy individuals and those with cardiovascular conditions. The dataset includes various types of ECG signals, providing the necessary variation for training our ensemble model effectively. To ensure the data is suitable for the machine learning process, several preprocessing techniques were applied, such as resizing the images to standard dimensions, converting them to grayscale, and enhancing features to improve the model's performance. The data collected from Mendeley was instrumental in developing and evaluating the cardiovascular disease prediction system.

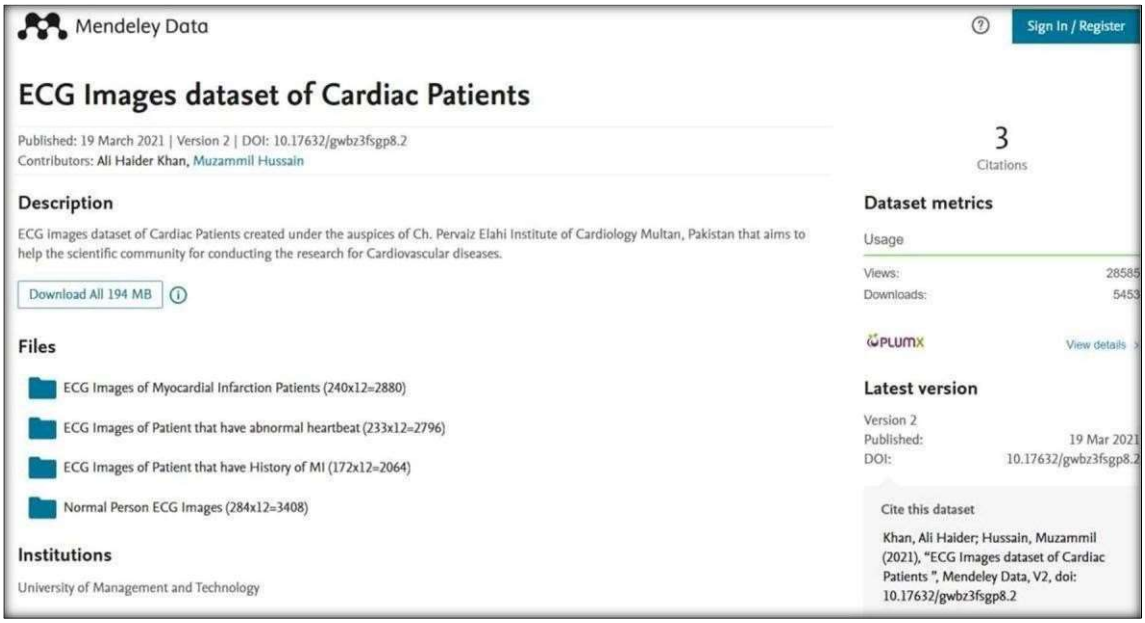

Figure 8.1: Experimental dataset sourced from Mendeley Data

This fig 8.1 showcases the source of the ECG dataset utilized in this study, obtained from the Mendeley Data repository. The dataset, titled "ECG Images Dataset of Cardiac Patients,"

8.2 SNAPSHOTS

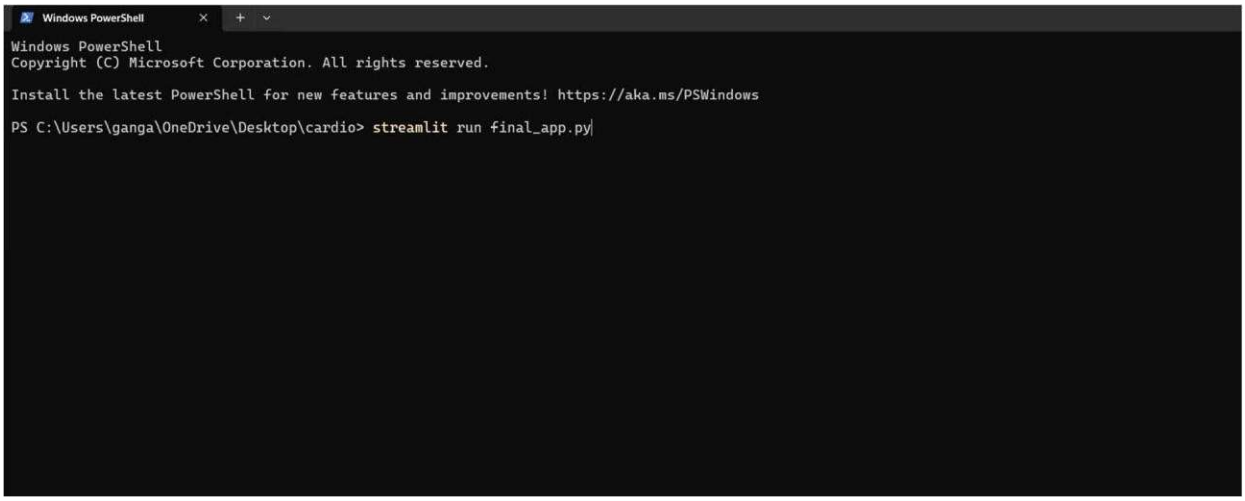

Figure 8.2: Starting Streamlit Application

Figure 8.2 shows the initial step of launching the Streamlit application from the terminal. The image displays the command executed to start the application, typically something like streamlit run app.py

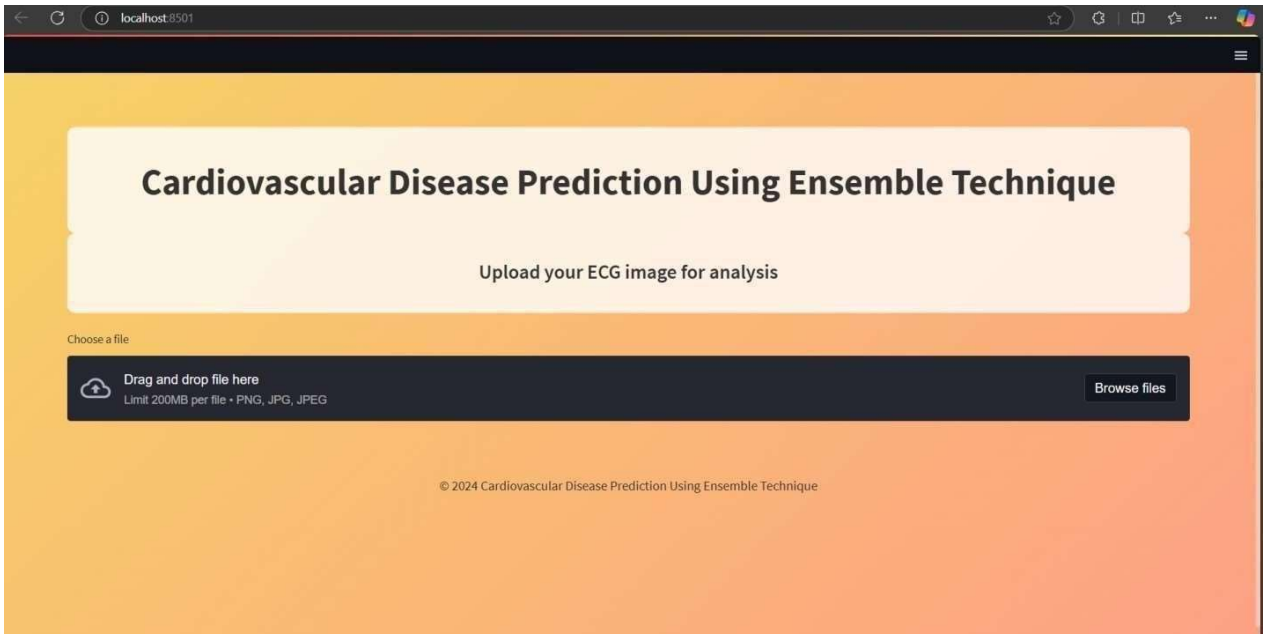

Figure 8.3: Dashboard of the project

Figure 8.3 presents the dashboard of the Cardiovascular Disease Prediction system's Streamlit application. The dashboard serves as the main interface where users can interact with the system.

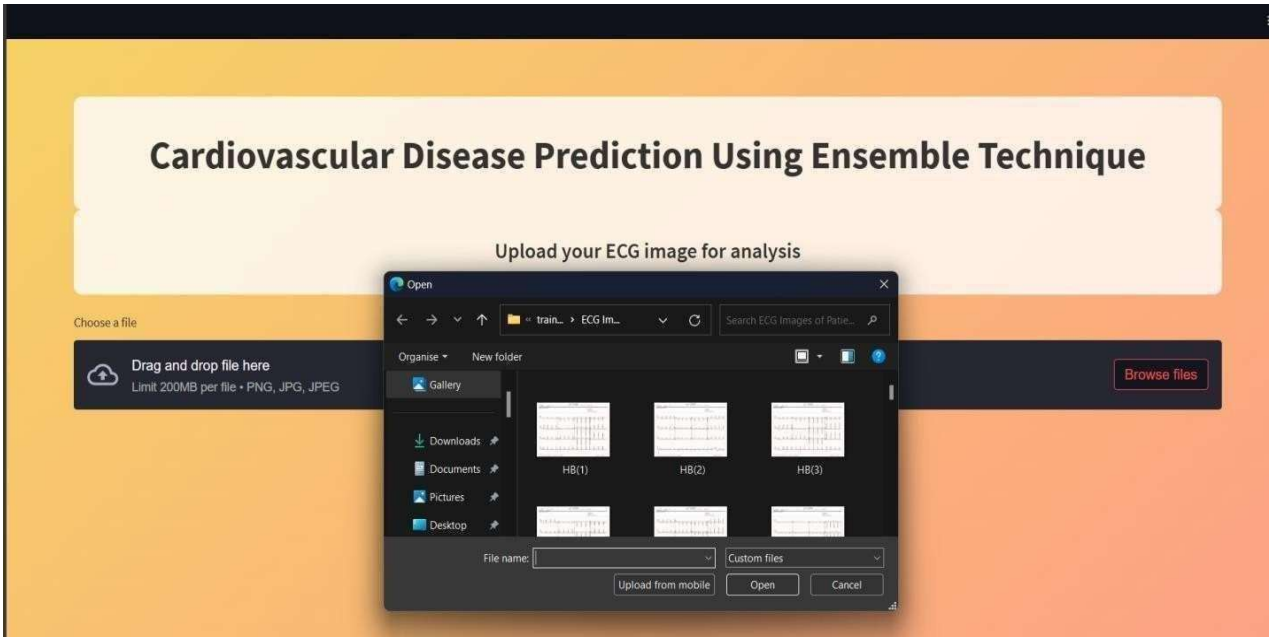

Figure 8.4: Uploading Image

Figure 8.4 illustrates the interface for uploading an ECG image within the Cardiovascular Disease Prediction system. This section of the application provides a clear, user-friendly widget where users can easily select and upload their image files.

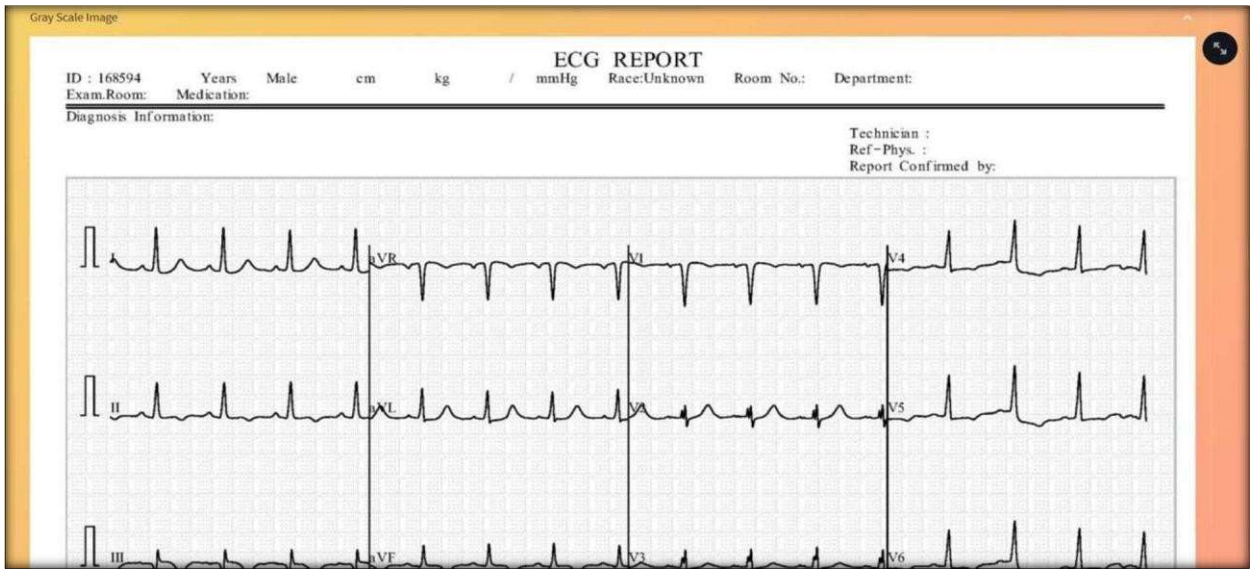

Figure 8.5: Gray Scale Image Obtained

Figure 8.5 displays the grayscale image obtained after preprocessing the uploaded ECG image. This transformation is a critical step in the image processing pipeline, where the original colored ECG image is converted into grayscale.

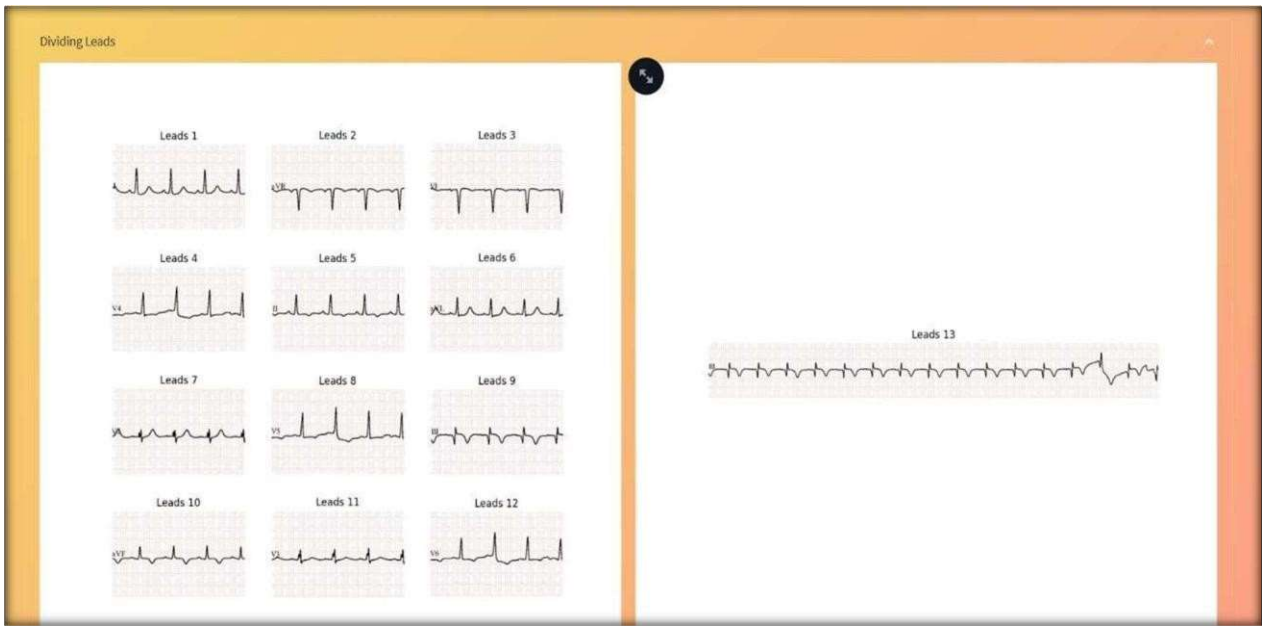

Figure 8.6: Division of ECG Leads

Figure 8.6 illustrates the division of ECG leads within the uploaded ECG image. In this step, the image is segmented into its respective leads, such as Lead I, Lead II, and Lead III, or other configurations depending on the type of ECG being processed.

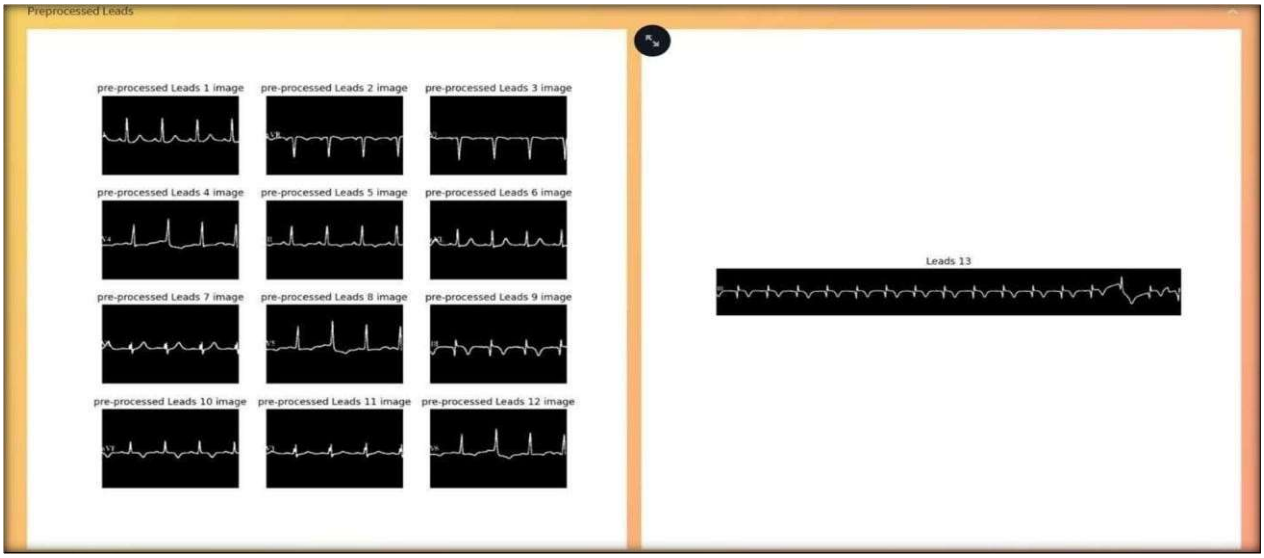

Figure 8.7: Preprocessed Leads

Figure 8.7 showcases the preprocessed leads extracted from the ECG image. After the image has been divided into individual leads, each lead undergoes a series of preprocessing steps such as noise reduction, normalization, and enhancement to improve the clarity and quality of the signals.

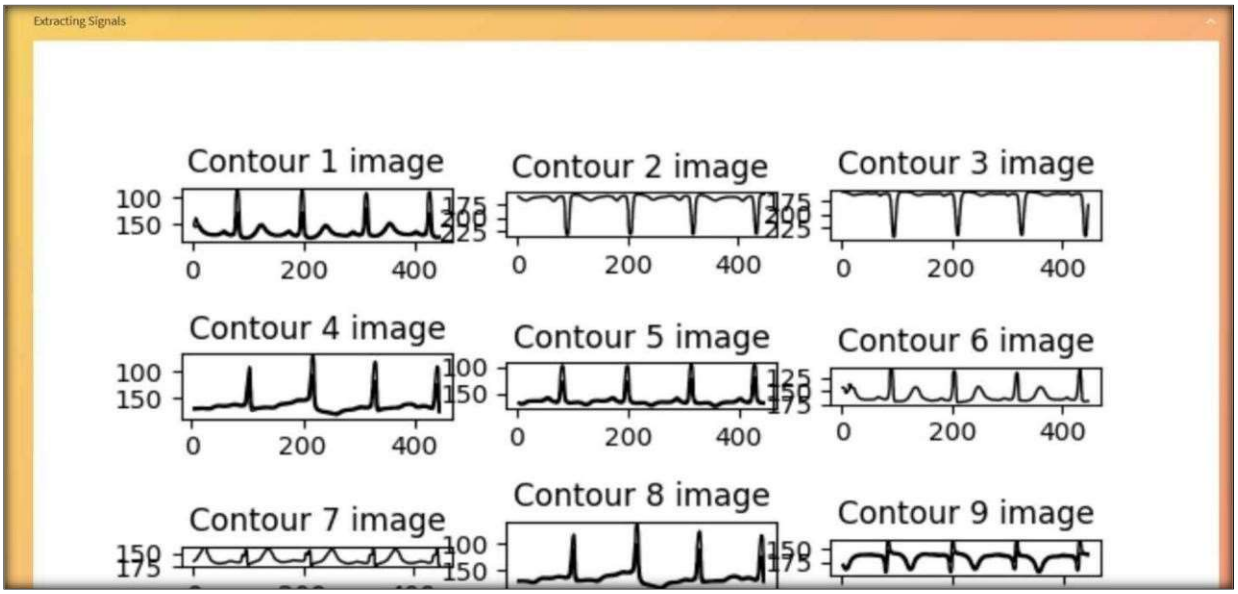

Figure 8.8: Contour Images Obtained

Figure 8.8 presents the contour images obtained from the preprocessed ECG leads. In this step, the contours of the ECG waveforms are extracted and visualized to highlight the critical features of the heart's electrical activity.

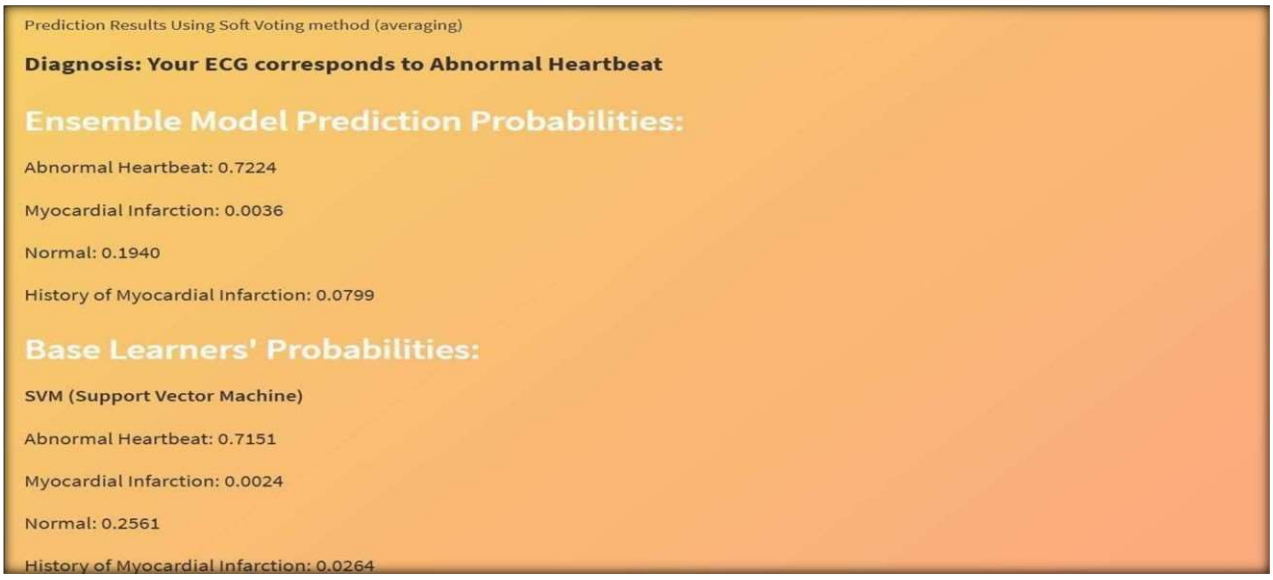

Figure 8.9: Predictive Probabilities for each class by ensemble and it's base learners (Soft Voting)

Figure 8.9 illustrates the predictive probabilities for each class generated by the ensemble model and its base learners using soft voting. In this step, the system combines the predictions from multiple base learners (such as decision trees, logistic regression, or support vector machines)

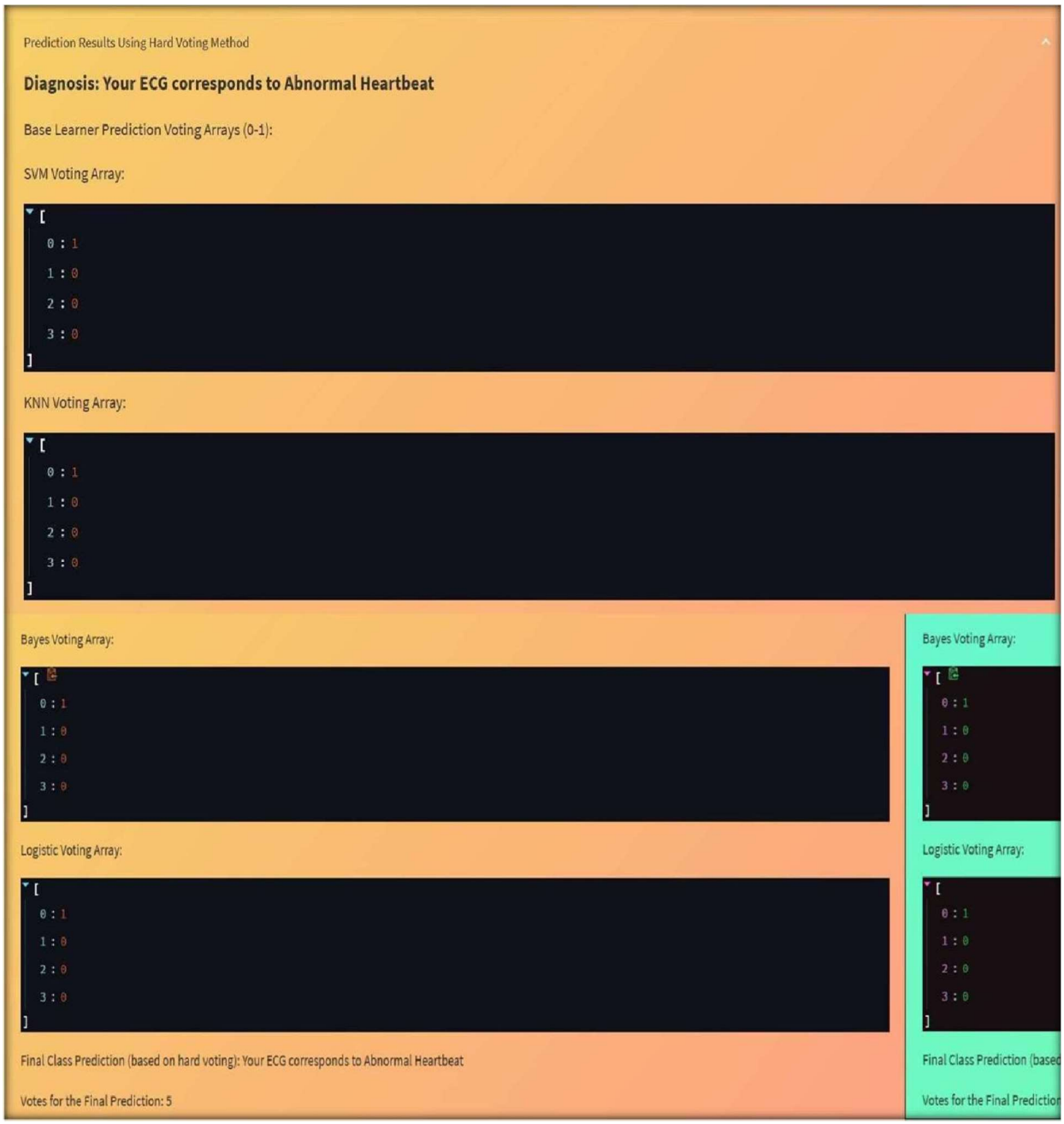

Figure 8.10: Predictive Probabilities for each class by ensemble and it's base learners (Hard Voting)

Figure 8.10 displays the predictive probabilities for each class generated by the ensemble model and its base learners using hard voting. Unlike soft voting, which averages the probabilities, hard voting combines the individual predictions of the base learners and selects the class that receives the majority vote. In this figure, the individual predictions of each base learner are shown, followed by the final predicted class determined by the ensemble model based on the majority vote.

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENTS

9.1 Conclusion

The project successfully demonstrates the power of ensemble machine learning algorithms in improving the accuracy and reliability of cardiovascular disease (CVD) prediction. By leveraging the strengths of multiple models, the system delivers enhanced predictive performance, overcoming the limitations of traditional diagnostic methods that often rely on single-model approaches. The integration of advanced image processing techniques with ensemble methods significantly improves the detection of subtle patterns in ECG signals, which is crucial for early diagnosis. Detecting these patterns enables timely clinical interventions, playing a vital role in preventing critical health events such as heart attacks and strokes.

Moreover, the automated nature of the system addresses the challenges associated with traditional CVD diagnosis, such as the heavy reliance on manual interpretation and the potential for human error. With its robust and scalable design, the system offers the flexibility to be deployed in a variety of clinical settings, particularly in resource-limited environments, where access to accurate and reliable diagnostic tools may be scarce. This enhances the overall accessibility and efficiency of healthcare, empowering clinicians to make data-driven decisions more quickly and accurately.

In addition to its immediate impact, this project represents a significant step forward in reducing the global burden of cardiovascular diseases by providing a reliable and efficient tool for diagnosis. As technology continues to advance, future improvements to the system—such as expanding datasets, refining machine learning algorithms, and incorporating additional patient features—will further enhance its performance and clinical applicability. These efforts will help solidify the system's role in revolutionizing cardiovascular disease diagnostics, offering both enhanced predictive capabilities and broader accessibility for healthcare providers worldwide. Furthermore, integration with electronic health records (EHR) could enable seamless adoption in clinical settings, improving workflow efficiency. The system's potential for real-time monitoring could also lead to earlier interventions, ultimately saving lives. Long-term, the system could contribute to the development of personalized healthcare approaches based on patient-specific data..

9.2 Future Enhancement

1. **Advanced Disease Detection and Classification:**

   Enhance the system to classify a broader range of cardiac diseases, such as myocardial infarction, arrhythmias, and congenital heart defects, using convolutional neural networks (CNNs). By integrating multi-modal imaging data like X-rays, CT scans, and MRIs, the system can provide more accurate and comprehensive diagnoses.

2. **Risk Stratification Models:**

   Develop CNN-based models for risk stratification by analyzing imaging data and clinical information to identify patterns linked to high-risk cardiovascular events. This feature can help prioritize patients requiring immediate intervention and personalized care plans.

3. **Automated Echocardiography Analysis:**

   Implement CNNs to automate the evaluation of echocardiograms for assessing cardiac function, valve morphology, and ejection fraction. This will ensure rapid and standardized analysis, reducing reliance on manual interpretation and enhancing diagnostic consistency.

4. **Future Event Prediction:**

   Extend the system to predict cardiovascular events by combining CNN outputs with clinical data, such as patient history, biomarkers, and lifestyle factors. This predictive model can guide preventive healthcare measures and proactive treatment strategies.

5. **Telemedicine and Remote Monitoring:**

   Leverage ensemble methods for real-time monitoring of cardiac health using wearable devices. The system can provide instant analysis of ECGs and other vital signs, enabling early detection of anomalies and remote healthcare consultations.

6. **Clinical Decision Support Systems (CDSS):**

   Integrate CNN-based predictions into clinical decision support systems to assist healthcare professionals. The system can provide real-time insights, second opinions, and data-driven recommendations for more informed and confident decision-making.

7. **Education and Training Tools:**

   Use the model as an educational tool by simulating diverse cardiac conditions for training

medical professionals. This enhancement can improve diagnostic skills and ensure practitioners are well-versed in interpreting complex cardiac imaging data.

8. **Quantitative Image Analysis:**
   Implement CNNs to perform detailed, quantitative analysis of cardiac imaging data. Metrics such as myocardial thickness, chamber volume, and blood flow can provide precise monitoring and progression tracking of cardiac conditions.
9. **Personalized Medicine:**
   Combine CNN-based outputs with genetic, environmental, and lifestyle data to create personalized treatment plans. This approach can optimize patient outcomes by tailoring interventions based on individual risk profiles and needs.
10. **Real-Time Multi-Disease Diagnosis:**
    Expand the system to simultaneously detect and diagnose multiple diseases, including comorbid conditions, by incorporating advanced multi-task learning techniques. This will make the model more versatile and applicable to a broader range of healthcare scenarios.

In terms of future enhancements, there are several areas where this system can be further improved to increase its efficiency, accuracy, and clinical applicability. One key area is the expansion of the dataset used to train the model. A more diverse and comprehensive dataset, including ECG images from different patient demographics, health conditions, and geographic regions, would improve the model's ability to generalize and handle a wider range of cardiovascular conditions. Additionally, integrating other forms of medical data, such as patient history, lab results, and demographic information, could provide a more complete picture of a patient’s health, leading to more personalized and accurate predictions.

Another significant enhancement involves refining the machine learning algorithms themselves. While ensemble methods have proven effective, exploring more advanced techniques, such as deep learning models (particularly Convolutional Neural Networks), could further elevate the system’s performance by enabling it to automatically detect more complex patterns in the ECG data. Moreover, optimizing the preprocessing and feature extraction steps would help improve the quality of the data fed into the model.

REFERENCES

[1] Tabassum, Tahmida, and Mohiuddin Ahmad. "Numerical data extraction from ECG paper recording using image processing technique." 2020 11th international conference on electrical and computer engineering (ICECE). IEEE, 2020.

[2] McAuliffe, Matthew J., et al. "Medical image processing, analysis and visualization in clinical research." Proceedings 14th IEEE symposium on computer-based medical systems. CBMS 2001. IEEE, 2001.

[3] Mahajan, Palak, et al. "Ensemble learning for disease prediction: A review." *Healthcare*. Vol. 11. No. 12. MDPI, 2023.

[4] Bauer, Eric, and Ron Kohavi. "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Their Variants." *Machine Learning*, vol. 36, no. 1-2, 1999, pp. 105–139. Springer, doi:10.1023/A:1007515423169

[5] Naderalvojoud, Behzad, and Tina Hernandez-Boussard. "Improving machine learning with ensemble learning on observational healthcare data." *AMIA Annual Symposium Proceedings*. Vol. 2023. 2024.