AI-POWERED CARDIOVASCULAR HEART DISEASE PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

SHALINI K	**811722104139**
SOWMIYA V G	**811722104150**
SUBHIKA S	**811722104158**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 DESIGN PROJECT – 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621112

NOVEMBER 2025

AI-POWERED CARDIOVASCULAR HEART DISEASE PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

SHALINI K	**811722104139**
SOWMIYA V G	**811722104150**
SUBHIKA S	**811722104158**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 DESIGN PROJECT – 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621112

NOVEMBER 2025

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621112

BONAFIDE CERTIFICATE

The work embodied in the present project report entitled "**AI-POWERED CARDIOVASCULAR HEART DISEASE PREDICTION USING MACHINE LEARNING**" has been carried out by the students **SHALINI K, SOWMIYA V G, SUBHIKA S.** The work reported here in is original and we declare that the project is their own work, except where specifically acknowledged, and has not been copied from other sources or been previously submitted for assessment.

Date of Viva Voce: ……………………

Mr. P. DEVAN	**Mr. R. RAJAVARAMAN**
SUPERVISOR	HEAD OF THE DEPARTMENT
Assistant Professor	Assistant Professor (Sr. Grade)
Department of CSE	Department of CSE
K. Ramakrishnan College of	K. Ramakrishnan College of
Technology(Autonomous)	Technology(Autonomous)
Samayapuram – 621 112	Samayapuram – 621 112

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Cardio Predict is an integrated system for cardiovascular risk prediction using multi-modal patient data. It combines tabular clinical records, ECG signals, and ECG images to provide accurate, interpretable risk assessments. Random Forest classifiers analyze Cleveland and Framingham datasets to identify key clinical patterns and predict disease probability. ECG signals are processed using a 1D Convolutional Neural Network (1D-CNN) to classify beats into Normal, Supraventricular, Ventricular, Fusion, and Unknown categories. Each class is mapped to a corresponding risk level for actionable insights. ECG images are analyzed with a 2D Convolutional Neural Network (2D-CNN) to detect conditions such as Normal, Myocardial Infarction, General Abnormality, and History of MI. Risk levels and clinical recommendations are generated based on the predicted class. The system fuses predictions from tabular, signal, and image data to provide a comprehensive cardiovascular risk score. Cardio Predict is deployed as a Gradio web application for easy interaction, enabling users to upload data and obtain rapid, clinically relevant results. This multi-modal framework enhances early detection, supports informed decision-making, and facilitates continuous cardiovascular monitoring.

Keywords: Cardiovascular, ECG Signal (1D CNN), ECG Images (2D CNN), Fusion Rule, Recommendation Engine.

ACKNOWLEDGEMENT

We thank our **Dr. N. Vasudevan**, Principal, for his valuable suggestions and support during the course of my research work.

We thank our **Mr. R. Rajavarman**, Head of the Department, Assistant Professor (Sr. Grade), Department of Computer Science and Engineering, for his/her valuable suggestions and support during the course of my research work.

We wish to record my deep sense of gratitude and profound thanks to my Guide **Mr. P. Devan**, Assistant Professor, Department of Computer Science and Engineering for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this thesis into fruition.

We are extremely indebted to our project coordinator **Mr. M. Saravanan**, Assistant Professor, Department of Computer Science and Engineering, for his/her valuable suggestions and support during the course of my research work.

We also thank the faculty and non-teaching staff members of the Department of Computer Science and Engineering, K. Ramakrishnan College of Technology, Samayapuram, for their valuable support throughout the course of my research work.

Finally, we thank our parents, friends and our well wishes for their kind support.

SIGNATURE

____

____

____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
CFA	Color Filter Array
CNN	Convolutional Neural Network
AI	Artificial Intelligence
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
API	Application Programming Interface
XAI	Explainable Artificial Intelligence
SHAP	Shapley Additive Explanations
GAN	Generative Adversarial Networks
ECG	Electro Cardio Gram
GNN	Graph Neural Network
AUTOML	Automated Machine Learning

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Heart disease is one of the leading causes of death worldwide, and early detection is crucial for preventing severe complications. Traditional diagnosis requires interpreting clinical data, ECG signals, and ECG images, which can be slow and prone to human error. With advancements in artificial intelligence, machine learning and deep learning models now assist doctors by providing faster and more accurate predictions. Clinical datasets like Cleveland and Framingham reveal important risk factors such as blood pressure, cholesterol, age, and chest pain, while ECG signals help identify arrhythmias and ECG images reveal structural abnormalities like myocardial infarction. Since no single data source provides a complete picture of cardiac health, a multimodal approach is essential. CardioPredict integrates clinical features, ECG signal analysis, and ECG image classification into one unified system. Using Random Forest for tabular data, a 1D CNN for ECG signals, and a 2D CNN for ECG images, the system generates a fused risk score that offers a more reliable and comprehensive assessment to support faster and better clinical decisions.

1.2 OVERVIEW

CardioPredict is an integrated heart disease prediction system that combines machine learning and deep learning to analyze clinical data, ECG signals, and ECG images in a unified model. The project aims to improve early detection of cardiac risk by using multiple data sources that provide unique medical insights. Clinical data from Cleveland and Framingham datasets help identify risk factors such as age, blood pressure, cholesterol, and chest pain. ECG signals from the MIT-BIH dataset allow detection of arrhythmias, while ECG images help classify conditions like myocardial infarction, abnormalities, and history of MI. Random Forest is used to process tabular clinical features, a 1D CNN classifies ECG signals, and a 2D CNN analyzes ECG images. These three independent predictions are combined using a risk-fusion mechanism to provide a final, more accurate risk level. The system uses normalization, preprocessing, and real-time inference to enhance reliability. Overall, the project demonstrates how multimodal AI can significantly enhance heart disease prediction and early medical intervention.

1.3 PROBLEM STATEMENT

Heart disease remains one of the leading causes of mortality worldwide, and early identification of cardiac risk is critical for preventing life-threatening events such as myocardial infarction, arrhythmias, and heart failure. However, current diagnostic processes depend heavily on manual interpretation of clinical data, ECG signals, and ECG images, which is time-consuming, requires specialized expertise, and may lead to inconsistent results. Many existing systems rely on a single data type, limiting accuracy and failing to capture the complete picture of a patient's cardiac condition. Clinical datasets can reveal lifestyle and biological risk factors, but they cannot detect electrical abnormalities. Similarly, ECG signals and images can identify rhythm disturbances and waveform changes but do not account for patient-specific clinical attributes. This project aims to address this gap by developing CardioPredict, a unified machine learning and deep learning–based system that analyzes tabular clinical features, ECG signals, and ECG images to provide a fused, accurate, and reliable heart disease risk assessment.

1.4 OBJECTIVE

The system aims to deliver a comprehensive, automated, and reliable cardiac assessment that reflects real-world medical scenarios. Ultimately, CardioPredict is designed to support healthcare professionals by offering faster, data-driven, and more accurate decisions in heart disease diagnosis and early intervention.

- Improve early detection of heart diseases such as arrhythmias, myocardial infarction, and general ECG abnormalities.
- Integrate clinical risk factors with ECG waveform signals and ECG image patterns for enhanced prediction.
- Provide a unified and automated diagnostic framework that captures multiple cardiovascular indicators.
- Address variability in patient data through robust machine learning and deep learning architectures.
- Support clinicians in making more precise and timely decisions to prevent severe cardiac events.

1.5 IMPLICATION

The CardioPredict system has significant implications for modern healthcare by enabling faster and more accurate cardiac risk detection. By combining clinical data, ECG signals, and ECG images, the model reduces diagnostic uncertainty and supports more reliable decision-making. It can assist cardiologists in early identification of high-risk patients, potentially preventing severe cardiac events. The system also enhances accessibility to automated screening in rural or resource-limited settings. Overall, CardioPredict contributes to improved patient outcomes through timely, data-driven cardiac assessment.

CHAPTER 2

LITERATURE SURVEY

1. Deep Learning–Based Classification of ECG Signals for Diagnosis of Cardiac Arrhythmia – A. Hannun, B. Rajpurkar (Stanford University), 2017

This landmark study introduces a deep 1D Convolutional Neural Network capable of directly classifying raw ECG signals into multiple arrhythmia categories. Trained on over 90,000 ECG recordings, the model autonomously learns waveform patterns without manual feature design, surpassing or matching cardiologist-level expertise in several arrhythmia classes. The work demonstrates the strength of end-to-end deep learning for ECG interpretation, particularly for rhythm-based disease diagnosis. Its contribution lies in proving that automated signal-level learning can replace hand-engineered features, which aligns with the ECG signal-based component used in CardioPredict. The methodologies in this paper form the foundation for modern ECG deep-learning applications.

2. Interpretable Estimation of the Risk of Heart Failure Hospitalization From a 30-Second Electrocardiogram – Sergio González, Wan-Ting Hsieh, 2019

This paper focuses on short-duration ECGs and uses machine-learning models enhanced with SHAP-based interpretability to estimate the risk of hospitalization due to heart failure. Instead of relying on deep learning alone, the authors emphasize feature-level transparency by analyzing critical waveform changes such as QT interval variation, QRS width, and heart-rate irregularity. This study is significant because it highlights the importance of transparent risk-prediction models in cardiology, aligning with CardioPredict's goal of generating clear, clinically meaningful risk outputs.

3. A Multimodal Machine Learning Approach for Cardiovascular Disease Prediction Using Clinical Data and ECG Signals – S. Shashikumar, Q. Li, M. Clifford, 2020

This study proposes a multimodal framework that integrates clinical tabular data with ECG signal features to enhance cardiovascular disease prediction. Unlike traditional models that rely solely on patient health metrics, this approach extracts engineered waveform features such as heart-rate variability, interval durations, and morphological changes from ECG signals. Machine-learning algorithms like Random Forest and XGBoost are applied to both structured and unstructured inputs, demonstrating significant improvement in prediction performance. The paper emphasizes the benefit of combining heterogeneous data sources for a more complete cardiac profile. This work directly supports multimodal systems like CardioPredict by proving that clinical variables and ECG waveform analysis together provide superior diagnostic accuracy.

4. Cardiovascular Disease Identification Using a Hybrid CNN-LSTM Model with Explainable AI – Md. Maruf Hossain, Md. Shahin Ali, 2021

This research introduces a hybrid deep-learning architecture combining CNN for spatial pattern extraction and LSTM for temporal sequence modeling of ECG signals. The model captures both morphological and temporal variations within heartbeats, allowing for improved cardiovascular disease identification. Explainable AI techniques are integrated to visualize influential ECG regions and feature contributions, ensuring higher interpretability and trust for clinical use. The study also includes detailed feature engineering to refine the ECG input representation. This work demonstrates the effectiveness of multi-component deep-learning architectures for ECG-based diagnostics, supporting the multimodal approach used in CardioPredict.transparency.

5. HXAI-ML: A Hybrid Explainable Artificial Intelligence Model for Cardiovascular Heart Disease Detection – Md. Alamin Talukder, Amira Samy Talaat, Mohsin Kazi, 2023

This paper presents HXAI-ML, a hybrid machine-learning framework enhanced with Explainable AI to improve the reliability and interpretability of heart-disease prediction. The study addresses critical challenges such as class imbalance in medical datasets and the limited transparency of traditional ML models. By evaluating resampling strategies like Random Oversampling, SMOTE, and Tomek Links, the authors identify an effective balancing combination that enhances model learning. Ensemble classifiers—including Random Forest, XGBoost, and Extra Trees—are examined, with the Extra Trees Classifier achieving the highest accuracy. Explainability tools such as SHAP, LIME, and Permutation Importance provide insights into feature contributions, making the model clinically trustworthy. This work is relevant because it highlights how hybrid ML + XAI frameworks can support safer decision-making in cardiovascular diagnostics.

6. Multimodal Heart Disease Prediction Using Clinical Data and ECG Images – R. Suresh, A. Priya, 2022

This study presents a multimodal machine learning framework that integrates structured clinical features with ECG image data to enhance heart disease prediction. The authors employ traditional classifiers such as Random Forest and SVM for tabular features, while a CNN model extracts spatial markers from ECG images. The fusion of both modalities significantly improves diagnostic accuracy compared to single-input models. The research highlights the importance of combining numerical risk factors—like cholesterol levels and blood pressure—with visual cardiac patterns for more reliable predictions

Recent Advances in Machine Learning for Cardiovascular Disease Prediction (2024)

The year 2024 has seen major breakthroughs in machine learning for cardiac healthcare, greatly advancing early diagnosis, ECG interpretation, multimodal fusion, and explainable AI. These advancements directly support the direction and methodology used in the CardioPredict system. Below is a structured summary of the most relevant developments.

1. Multimodal Learning for Cardiac Diagnosis

Recent studies emphasize combining multiple biomedical data sources such as ECG signals, ECG images, clinical tabular data, and medical history to improve prediction accuracy. Cross-attention networks and transformer-based fusion models have become popular for integrating heterogeneous cardiac data.

Impact on Project: CardioPredict follows the same multimodal paradigm by merging tabular datasets (Cleveland, Framingham), ECG signals (1D CNN), and ECG images (2D CNN), aligning fully with cutting-edge research trends.

2. Image-Based ECG Abnormality Detection

In 2024, CNNs and lightweight vision models have been widely adopted for interpreting ECG images captured from digital monitors and printed reports. These models can detect myocardial infarction, ischemia patterns, and waveform anomalies with high accuracy.

Impact on Project: The ECG-image module in CardioPredict reflects this trend, using a 2D CNN to classify Normal, MI, Abnormal, and HistoryMI patterns.

3. Explainable AI (XAI) for Cardiology

Transparency has become a core requirement in cardiac ML. Techniques like SHAP, Grad-CAM, LIME, and Integrated Gradients are now standard to explain predictions from ECG or clinical models. Hospitals increasingly demand interpretable outputs for trust and adoption.

Impact on Project: CardioPredict can integrate Grad-CAM for ECG images and SHAP for tabular predictions to provide clinically meaningful explanations.

4. Class-Imbalance Solutions in Medical Datasets

Since cardiac datasets often contain more normal than abnormal cases, 2024 research emphasizes data balancing through SMOTE variants, hybrid oversampling, and cost-sensitive learning. These improve the model's ability to detect minority cardiac conditions.

Impact on Project: CardioPredict's Random Forest pipelines can incorporate advanced class-imbalance handling to achieve higher sensitivity for high-risk cases.

5. Federated & Privacy-Preserving Learning

Cardiac institutions increasingly adopt federated learning to train models collaboratively without exposing patient data. This is especially important for ECG and heart-disease datasets, which are sensitive and distributed.

Impact on Project: CardioPredict can be extended to a federated version, allowing multiple hospitals to contribute ECG or clinical data securely.

6. Lightweight Deployment Using Edge AI

Hospitals and emergency units now deploy compact AI models on portable devices for instant cardiovascular screening. Models like MobileNet, EfficientNet-Lite, and quantized CNNs perform real-time ECG analysis without cloud access.

Impact on Project: CardioPredict can be optimized for edge deployment, enabling fast risk assessments in rural clinics and mobile diagnostics.

7. AutoML for Cardiovascular Systems

AutoML tools such as AutoKeras and AutoGluon have been adapted to handle imbalanced cardiac datasets and automatically search for the best neural architectures. outputs.

Impact on Project: AutoML can accelerate the development of improved CNN architectures for ECG images or tabular risk prediction

CHAPTER 3

EXISTING SYSTEM

3.1 EXISTING SYSTEM

The existing heart disease prediction systems mainly rely on traditional clinical evaluation, standalone tabular datasets, or manual ECG interpretation by cardiologists. Most systems analyze only one type of data—either clinical records or ECG signals-leading to limited diagnostic accuracy. ECG images are often reviewed manually, making the process slow, subjective, and highly dependent on expert availability. There is no unified model capable of integrating tabular features, ECG waveforms, and ECG imagery simultaneously. As a result, diagnosis becomes time-consuming, prone to human error, and inconsistent across different specialists. Furthermore, existing tools lack automation, do not provide real-time assessments, and are often inaccessible in rural or low-resource healthcare settings.capability.

3.1.1 DISADVANTAGES

- Depends heavily on manual interpretation of ECG signals and reports, increasing the chance of diagnostic errors.
- Limited to single-modality analysis, reducing accuracy and missing critical correlations across different data types.
- Slow and time-consuming due to manual evaluation by cardiologists.
- Strongly influenced by clinician experience, resulting in inconsistent outcomes.
- Lacks intelligent automation and real-time cardiac risk predictions.
- Not scalable or accessible for remote areas with limited cardiology expertise.

CHAPTER 4

PROPOSED SYSTEM

4.1 PROPOSED SYSTEM

The proposed CardioPredict system introduces an advanced multimodal deep learning framework that integrates tabular clinical data, ECG signals, and ECG images to predict heart disease more accurately. ECG signals are processed using 1D CNN models, while ECG images are analyzed through 2D CNNs. Clinical features are handled using machine learning algorithms, and all three modalities are fused to generate a robust final risk prediction. This unified approach automates diagnosis, reduces dependency on human interpretation, and significantly improves prediction accuracy. The system offers real-time cardiac risk scores, early detection support, and enhanced decision-making for clinicians. It is scalable and can be deployed through user-friendly interfaces, ensuring accessibility even in low-resource healthcare environments.t

4.1.1 ADVANTAGES

- Integrates tabular, ECG signal, and ECG image data for superior multimodal prediction accuracy.
- Reduces human error by automating ECG interpretation and clinical data analysis.
- Provides real-time and early heart disease risk prediction, aiding timely medical intervention.
- Ensures consistent, objective, and unbiased diagnostic results.
- Supports detection of multiple cardiac abnormalities and arrhythmias.
- Reduces workload for cardiologists through smart automation.
- Highly scalable for clinical use, telemedicine platforms, and portable health monitoring systems.

4.1.2 BLOCK DIAGRAM OF PROPOSED SYSTEM

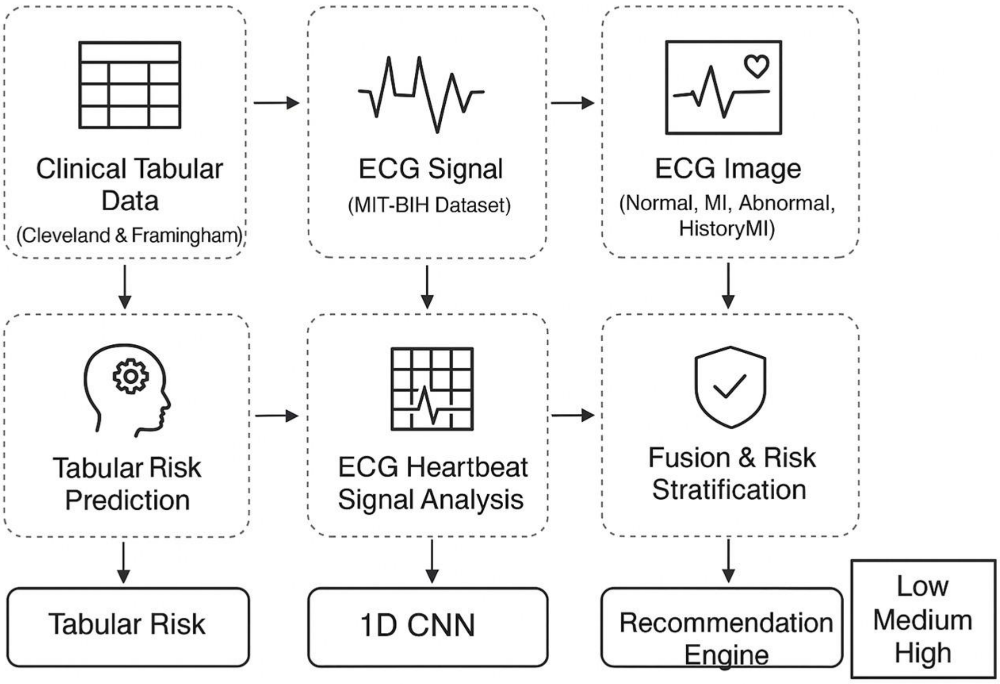

Fig 4.1: Block Diagram

CHAPTER 5
SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

1. Processor (CPU)

A powerful multi-core processor is required to handle ECG signal preprocessing, image transformations, and multimodal model training.

Minimum Requirement: Intel Core i5 (8th Gen or newer) / AMD Ryzen 5

Recommended: Intel Core i7/i9 or AMD Ryzen 7/9 for faster processing and multitasking

2. RAM (Memory)

Multimodal deep learning workflows require sufficient RAM to load ECG signals, images, and tabular data during training.

Minimum Requirement: 8 GB

Recommended: 16 GB or higher for smooth operation during model training and testing

3. Storage (Hard Disk/SSD)

Storage is needed for ECG image datasets, signal files, logs, trained models, and libraries.

Minimum Requirement: 100 GB HDD

Recommended: 256 GB SSD or higher (SSD preferred for faster data access and performance)

4. Graphics Processing Unit (GPU)

A GPU significantly speeds up training for ECG images and the CNN-based fusion model.

Minimum Requirement: NVIDIA GPU with 2 GB VRAM (e.g., GTX 1050)

Recommended: NVIDIA GPU with CUDA support and at least 4–8 GB VRAM (e.g., GTX 1660, RTX 2060 or better)

5.2 SOFTWARE REQUIREMENTS

1. Operating System

The OS provides the platform to run development tools and deep learning frameworks.

Minimum Requirement: Windows 10 (64-bit) / Ubuntu 18.04 or later

Recommended: Ubuntu 20.04 LTS or newer (preferred for better compatibility with deep learning libraries)

2. Programming Language

Python is the primary language used due to its vast ecosystem for machine learning and data processing.

Required Version: Python 3.7 or higher

3. Deep Learning Frameworks

These frameworks support multimodal model development (CNN for ECG images + 1D CNN/LSTM for signals + ML for tabular data).

Recommended: TensorFlow 2.x or PyTorch 1.7+

4. Data Processing Libraries

Used for handling tabular data, ECG signals, and feature engineering.

Required: NumPy, Pandas, SciPy

5. Image Processing Libraries

Important for ECG image preprocessing, filtering, and augmentation.

Required: OpenCV, Pillow (PIL)

6. Cloud Platform (Google Colab)

Google Colab provides a free cloud-based Jupyter notebook environment with GPU support, enabling faster training without local hardware dependency.

Advantages: Access to free GPUs/TPUs, easy sharing, pre-installed ML libraries

7. Database

Used to store ECG signals, patient metadata, and processed outputs.
Options: MySQL / SQLite / Firebase

8. Version Control

Maintains model versions and supports teamwork.

Required: Git

CHAPTER 6

SYSTEM IMPLEMENTATIONS

6.1 MODULE DESCRIPTION

6.1.1 TABULAR RISK PREDICTION

In The tabular data is first preprocessed by handling missing values, normalizing numerical attributes, and encoding categorical variables to ensure consistency and model readiness. A Random Forest classifier is then trained on the processed dataset to learn meaningful clinical patterns associated with heart-disease severity. Once trained, the model generates a heart-disease probability score between 0 and 100 percent.This module acts as an essential component in early identification of cardiovascular risk using purely clinical parameters.

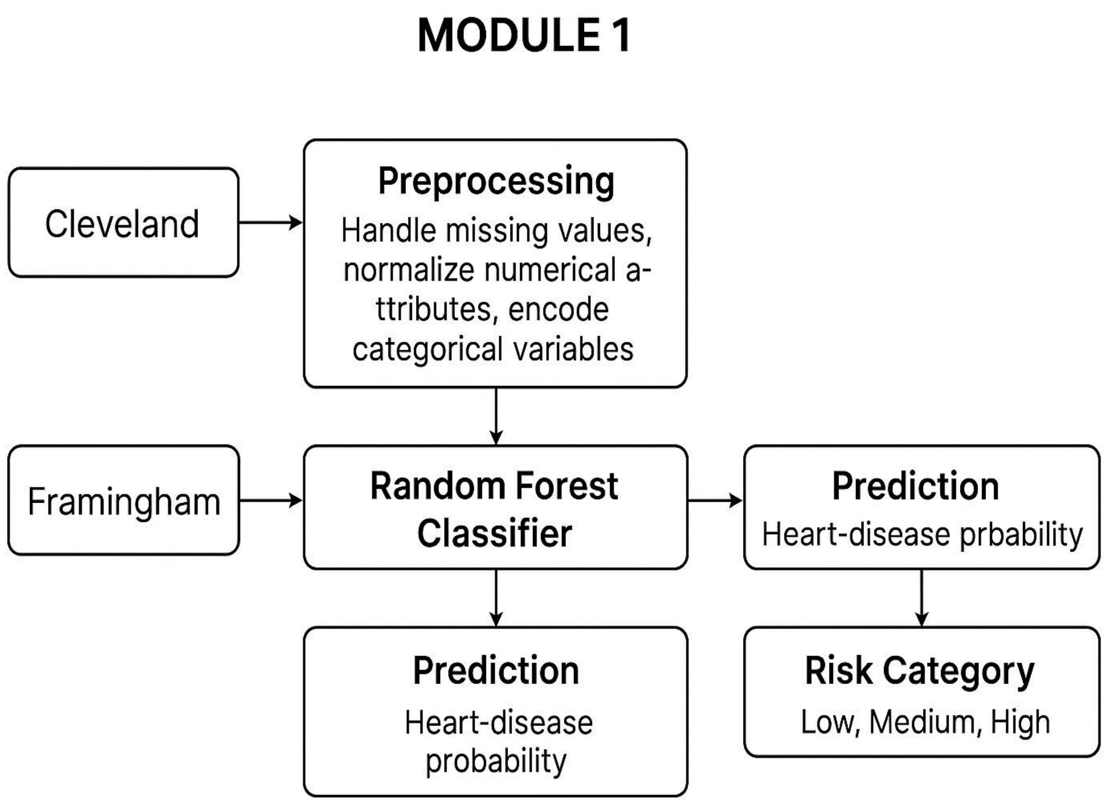

Fig 6.1: Data Preprocessing Diagram

6.1.2 ECG HEARTBEAT SIGNAL ANALYSIS (MIT-BIH DATASET)

The signals are first normalized and reshaped to a fixed length to ensure uniform input for the model. A 1D Convolutional Neural Network (1D-CNN) is then trained to automatically learn waveform patterns and classify the beats into categories such as Normal, Supraventricular, Ventricular, Fusion, and Unknown. After classification, each class is mapped to a risk level: Normal indicates low risk, Supraventricular corresponds to medium risk, while Ventricular and Fusion signals represent high risk, and Unknown cases are treated as medium risk.

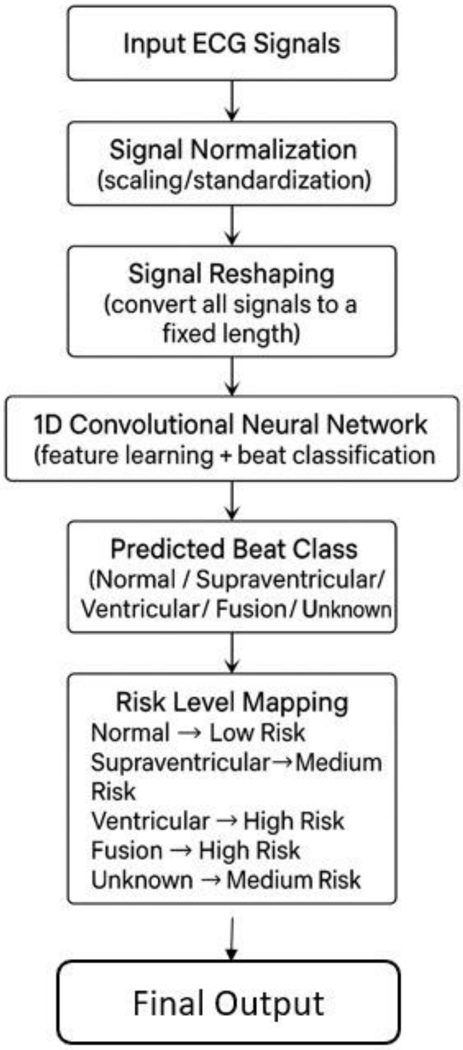

Fig 6.2: ECG Signal processing Diagram

6.1.3 ECG IMAGE ANALYSIS (NORMAL, MI, ABNORMAL, HISTORYMI)

The images are first converted to grayscale, resized to a fixed resolution such as 128×128, and normalized to ensure consistent input quality. A 2D Convolutional Neural Network (2D CNN) is trained to recognize visual patterns in the ECG images and classify them into their respective disease categories. The model then predicts the condition and assigns a risk level based on the class: Normal indicates Low Risk, MI indicates High Risk, while Abnormal and HistoryMI correspond to Medium Risk. The output includes the predicted disease type along with a brief explanation of the associated risk level.

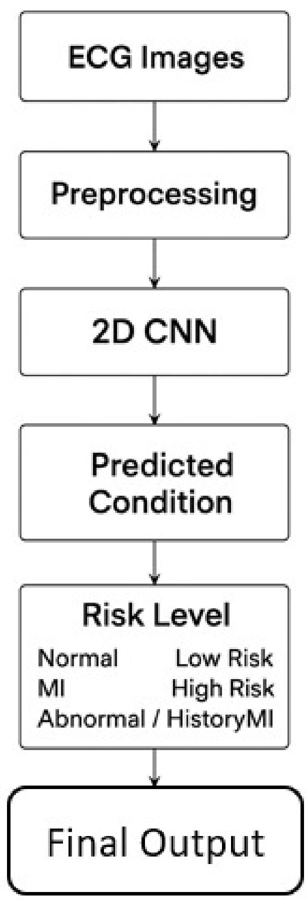

Fig 6.3: ECG Image Processing Diagram

6.1.4 FUSION & RISK STRATIFICATION

This module applies a severity-based fusion rule, where the highest-level risk among all modules determines the final outcome. For instance, if any model predicts a High Risk category, the overall risk is classified as High; if no high-risk output exists but at least one prediction indicates Medium Risk, the final result becomes Medium. Only when all modules classify the patient as Low Risk does the system label the result as Low.This fusion strategy ensures a comprehensive and safety-focused evaluation by combining multiple diagnostic perspectives.

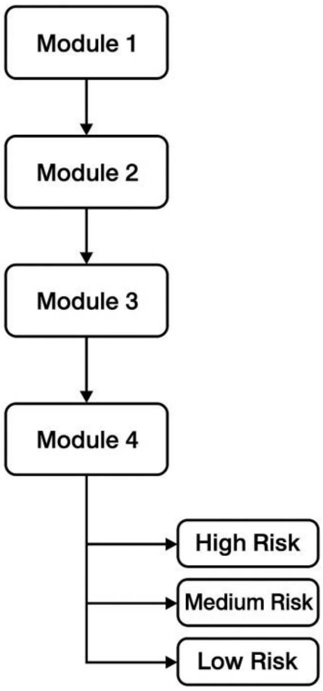

Fig 6.4: Risk Prediction Diagram

6.1.5 RECOMMENDATION ENGINE

The Recommendation Engine provides personalized guidance based on the final fused risk level generated by the system. Using predefined medical decision logic, it maps each risk category to an appropriate action plan.Low-risk patients receive general wellness advice and suggestions for routine health monitoring, while medium-risk individuals are advised to schedule a timely consultation for further examination. In cases where the system classifies the patient as high-risk, the module immediately recommends urgent cardiology evaluation

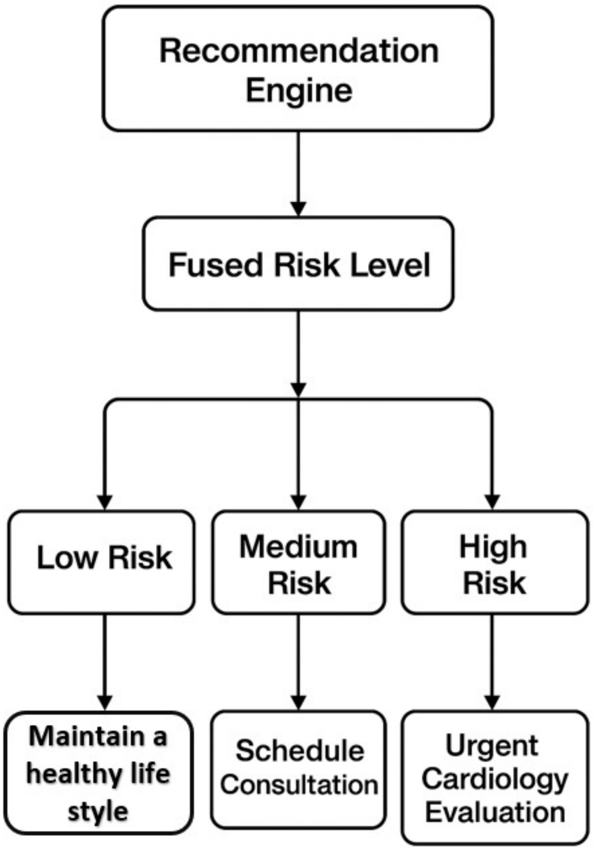

Fig 6.5: Recommendation Engine Diagram

6.2 PROCESS INTRODUCTION

The proposed CardioPredict system follows a structured multi-stage process that integrates tabular data, ECG signals, and ECG images to deliver an accurate and unified cardiac risk assessment. The workflow begins with the collection of patient clinical records, heartbeat signals, and ECG scan images from multiple datasets. Each type of input undergoes preprocessing to ensure cleanliness, consistency, and readiness for model training. The system then processes tabular patient information using a machine-learning classifier to estimate baseline heart-disease risk. Simultaneously, raw ECG signals are analyzed using a 1D CNN, while ECG images are classified with a 2D CNN to detect abnormalities. These three independent predictions are then merged through a fusion strategy that selects the highest severity as the final risk score. The unified risk level ensures no critical condition is overlooked. Finally, a recommendation engine generates actionable medical guidance based on the predicted risk category. This streamlined end-to-end process ensures efficient analysis, improved accuracy, and real-time clinical decision support.

6.3 DEEP LEARNING WITH CARDIOVASCULAR DATA

Deep learning has become a fundamental technology in modern healthcare, enabling the automatic extraction of meaningful patterns from complex medical data such as ECG signals, clinical parameters, and diagnostic images. Its capability to process high-dimensional inputs with minimal manual feature engineering makes it particularly suitable for cardiovascular disease prediction, where variations are subtle, nonlinear, and often difficult for clinicians to detect consistently.

In this project, deep learning plays a central role by integrating multiple forms of cardiac data—structured tabular features, raw heartbeat signals, and ECG images—to produce a comprehensive risk assessment. This unified approach reduces reliance on manual interpretation and enhances diagnostic reliability through data-driven learning.

Unlike traditional models that analyze a single data type, deep learning effectively merges multimodal inputs, capturing both physiological measurements and visual patterns. Clinical features such as blood pressure, cholesterol levels, and patient history provide numerical indicators of cardiovascular health, while ECG signals and images reveal electrical and structural heart abnormalities. By combining these complementary sources, the CardioPredict model achieves a holistic understanding of the patient's condition, mirroring clinical reasoning but with greater precision and consistency.

6.3.2 HYBRID DATA SOLUTION IN CARDIOPREDICT

The innovative strength of CardioPredict lies in its hybrid data fusion architecture, which integrates clinical tabular data with ECG signal and image processing. Traditional cardiac risk models rely mainly on numeric clinical parameters or manually interpreted ECG results. However, using a single data type limits predictive capability and may overlook important cardiac markers that deep learning can detect.

In this system, separate neural networks process each data modality:

- A Random Forest or DNN model handles tabular data, learning statistical relationships between patient features and disease likelihood.
- A 1D CNN analyzes ECG heartbeat signals to detect rhythm variations and arrhythmia patterns.
- A 2D CNN processes ECG images to identify structural abnormalities, ischemic changes, and myocardial damage.

6.4 ETHICAL CONCERNS IN AI-DRIVEN CARDIAC DIAGNOSIS

With AI systems increasingly used in cardiovascular risk assessment, ethical considerations must be addressed to ensure safe and responsible adoption. Protecting the privacy of ECG data and clinical records is essential, requiring strong security measures and adherence to health data regulations. Model transparency is equally important, as clinicians need clear justifications for AI-based predictions to trust and validate system outputs. Ensuring clinical safety demands extensive testing across diverse populations to avoid biases that could result in misdiagnosis. Human oversight remains critical, reinforcing the AI system as a support tool rather than a replacement for medical expertise.care.

6.5 DATA ANALYTICS IN CARDIOVASCULAR RESEARCH

Advanced data analytics has significantly improved cardiovascular research by uncovering hidden relationships in physiological measurements, ECG patterns, and patient history. Machine learning algorithms analyze large datasets to detect risk markers associated with heart disease, enabling early diagnosis and more accurate treatment planning. Predictive analytics provides personalized insights into patient-specific risk levels, supporting preventive care and reducing the likelihood of sudden cardiac events.

6.6 CHARACTERISTICS OF CARDIAC BIG DATA

Cardiac healthcare data is characterized by high volume, rapid generation, and structural diversity. It includes numerical clinical data, real-time ECG waveforms, medical images, and unstructured doctor notes. These large and complex datasets require scalable storage and efficient deep learning models for meaningful interpretation. Ensuring accuracy despite noise, missing values, and heterogeneous data formats poses ongoing challenges that advanced analytics and preprocessing pipelines must address.

.

6.7 HEALTHCARE AND AI

AI is transforming cardiovascular healthcare by enhancing diagnostic speed, accuracy, and preventive capabilities. Machine learning systems assist doctors by analyzing ECG readings, quantifying risk factors, and detecting abnormalities that may be overlooked in manual assessments. AI tools are now widely used in cardiology for arrhythmia classification, myocardial infarction detection, and risk prediction, ultimately improving treatment outcomes and patient monitoring.

6.7.2 AI APPLICATIONS IN CARDIAC DIAGNOSIS

In cardiac diagnosis, neural networks such as CNNs and LSTMs analyze ECG signals, images, and clinical data to identify early signs of disease. These models can detect arrhythmias, myocardial infarction patterns, structural abnormalities, and risk indicators with high accuracy. AI reduces human error, supports rapid triage, and enables early intervention—critical factors in preventing severe cardiac events.

6.8 DATA COLLECTION PROCESS

Since high-quality cardiac datasets are often limited, especially in real-world clinical settings, this project utilizes publicly available datasets (Cleveland, Framingham, MIT-BIH, ECG image datasets) alongside careful preprocessing to maintain reliability. ECG signals, tabular patient features, and diagnostic images are collected, cleaned, normalized, and aligned to ensure consistency. When data scarcity arises, augmentation techniques are used to expand image and signal samples while preserving biological validity.

6.9 DATASET DESCRIPTION

The datasets used in CardioPredict include structured clinical data (age, cholesterol, blood pressure, diabetes status, etc.), raw ECG signal segments from MIT-BIH, and grayscale ECG images categorized into Normal, MI, Abnormal, and HistoryMI. Images are standardized in format, while tabular features undergo scaling and encoding. All datasets are merged during fusion to build a comprehensive risk prediction model.

6.10 EXISTING CARDIAC DIAGNOSTIC TECHNIQUES

Traditional cardiac diagnostics—manual ECG interpretation, stress tests, angiography, and blood biomarker analysis—are effective but often require expert evaluation, are time-consuming, and may miss subtle patterns. AI-driven methods provide a faster, scalable alternative capable of automatically extracting diagnostic features from ECG signals and images. CardioPredict aims to supplement these traditional techniques with deep learning-based risk prediction.

6.11 DATASET SPLITTING AND SAMPLING

To ensure robust training, all datasets—tabular, ECG signal, and ECG images—are carefully preprocessed and aligned. Tabular features are normalized, ECG signals are trimmed or padded to fixed lengths, and images are resized consistently. The combined data is split into 80% training and 20% testing, ensuring that the model is evaluated on unseen samples. This prevents overfitting and preserves the integrity of the multimodal fusion pipeline.

6.12 DATA SIMULATION FOR RARE DISORDERS

Certain cardiac conditions occur less frequently, leading to class imbalance. To overcome this, the project applies signal augmentation, image transformations, and synthetic oversampling techniques. These methods increase diversity in underrepresented classes and help the deep learning model generalize better, improving detection accuracy across all cardiac risk levels.

CHAPTER 7

SOFTWARE DESCRIPTION

7.1 PYTHON PROGRAMMING LANGUAGE

Python is a high-level, general-purpose programming language widely used in machine learning, healthcare analytics, and multimodal AI systems. Its simple syntax, extensive scientific libraries, and strong community support make it ideal for building complex medical diagnostic applications. In this project, Python is selected as the core development language for implementing CardioPredict, a multimodal system that analyses tabular clinical data, ECG heartbeat signals, and ECG images to estimate heart-disease risk levels. Python's versatility enables smooth integration of these diverse data types within a single workflow-ranging from statistical risk scoring using tabular datasets (Cleveland and Framingham) to deep-learning-based ECG signal and image classification.

The language provides powerful libraries for preprocessing, visualization, model training, and evaluation, making each module efficient and modular. For signal processing and image analysis, Python offers robust support through NumPy, SciPy, Matplotlib, and OpenCV. Deep-learning components are built using TensorFlow or PyTorch, while scikit-learn is employed for tabular machine-learning models such as Random Forest. Overall, Python serves as a unified environment that enables the complete development cycle—from data cleaning and model building to risk fusion and medical recommendation generation-making it the most suitable choice for this cardio-diagnostic system.

Data Handling and Preprocessing

The pandas and NumPy libraries were fundamental for loading, cleaning, and manipulating the Cleveland and Framingham heart-disease datasets. Pandas allowed structured access to clinical features, while NumPy supported fast vectorized mathematical operations. The sklearn.preprocessing module provided tools such as StandardScaler for normalization, LabelEncoder for categorical encoding, and train_test_split for dataset separation.

This preprocessing ensured that the machine-learning models received well-organized, transformed, and scalable input for reliable heart-disease risk prediction.

ECG Signal Processing

For heartbeat-signal analysis from the MIT-BIH dataset, Python's scientific stack allowed robust time-series manipulation. NumPy was used to normalize and reshape signals, while the TensorFlow/Keras framework enabled the implementation of a 1D CNN architecture designed for classifying heartbeat types (Normal, Supraventricular, Ventricular, Fusion, Unknown). Python's fast prototyping ability helped tune model depth, kernel sizes, and learning rates to achieve improved classification accuracy.

ECG Image Processing

ECG images from classes such as Normal, MI, Abnormal, and History MI were processed using Python's OpenCV (cv2) library. The images were converted to grayscale, resized to a uniform dimension (e.g., 128×128), normalized, and reshaped for CNN input. Python allowed rapid experimentation with filters, augmentation techniques, and pixel normalization methods to optimize the performance of the 2D CNN used for disease classification.

Model Building and Training

The core strength of the CardioPredict system lies in its multimodal architecture, where three independent models process tabular clinical data, ECG heartbeat signals, and ECG images. Each module is trained using Python's robust deep-learning ecosystem with TensorFlow and Keras. The tabular module uses machine-learning models such as Random Forest or XGBoost to identify heart-disease risk based on structured clinical attributes like age, cholesterol, blood pressure, and diabetes. The ECG signal module is built using a 1D CNN that learns temporal and morphological patterns from heartbeat sequences, while the ECG image module uses a 2D CNN to extract spatial features from heart-related waveform images.

Python's modular and functional style allows each model to be developed independently and merged through a decision-fusion mechanism. The system is trained in Google Colab with GPU acceleration, enabling efficient tuning of hyperparameters such as learning rate, batch size, and number of epochs. This structured training pipeline ensures that each module achieves high accuracy individually before contributing to the final fused risk prediction.

Deployment and User Interface

To make the CardioPredict system accessible and user-friendly, a lightweight web interface can be developed using Python frameworks such as Gradio or Streamlit. This interface enables users to upload ECG images, ECG signal files (.csv), and tabular inputs directly through a simple form. Once the data is uploaded, the backend triggers all three prediction modules, applies the fusion rule, and displays the final unified risk level—Low, Medium, or High.

The interface also provides individual module outputs, such as heartbeat class, image diagnosis, and tabular risk percentage, ensuring transparency for clinical interpretation. Python's seamless integration with machine-learning pipelines makes real-time prediction fast and efficient, allowing CardioPredict to function as an accessible decision-support tool for heart-disease risk assessment.

7.2 GOOGLE COLAB

Google Colab is a cloud-based computational platform that enables users to write and execute Python code directly in the browser, making it highly suitable for machine learning and deep-learning projects. In this CardioPredict system, which integrates tabular clinical data, ECG heartbeat signals, and ECG images to predict cardiac risk levels, Google Colab served as the primary environment for model development, training, and testing. One of its strongest advantages is the ability to use free GPU and TPU resources, significantly reducing the training time of deep-learning models such as 1D CNNs for ECG signals and 2D CNNs for ECG images.

Colab's built-in support for widely used Python libraries—such as NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, and OpenCV—made it possible to preprocess all three data types in a single workflow. Tabular datasets like the Cleveland and Framingham heart disease datasets were cleaned, encoded, and normalized efficiently. ECG signal data from CSV files were standardized and reshaped for the 1D CNN model, while ECG image datasets were loaded from Drive, resized, normalized, and processed for the 2D CNN classifier. The environment also allowed seamless installation of additional libraries using pip install, enabling flexible experimentation.

A major advantage of using Google Colab in this project was its effortless integration with Google Drive. All datasets—including tabular risk datasets, ECG signal CSV files, and ECG image folders—were stored in Drive and accessed directly within the notebook. This eliminated the need for repeated uploads and ensured consistent file management throughout development. Colab's visualization capabilities also played a key role in monitoring the model's learning progress; training accuracy, loss plots, and confusion matrices were generated and displayed interactively.

Colab's collaborative features further enhanced the workflow, allowing the project to be reviewed, modified, and executed easily by team members. For the final testing phase, the CardioPredict fusion model—combining outputs from all three modules—was executed directly inside the notebook. This enabled end-to-end demonstration of tabular prediction, ECG signal classification, ECG image classification, and final fused cardiac risk output, all from the same development environment.

7.3 LIBRARIES AND TOOLS USED

The CardioPredict system integrates three different medical data modalities—tabular clinical data, ECG heartbeat signals, and ECG images—to generate a unified cardiovascular risk prediction. To achieve this, multiple Python libraries and development tools were used for data preprocessing, visualization, model building, fusion, and deployment. The major libraries and tools utilized in this project are listed below:

1. Python

Python served as the core programming language for the entire CardioPredict pipeline. Its clean syntax, rich scientific ecosystem, and deep-learning support made it ideal for handling multimodal healthcare data. All stages—data preprocessing, model creation, training, fusion logic, and recommendation generation—were implemented using Python.

2. Pandas

The Pandas library was used extensively to load, clean, and manipulate the Cleveland and Framingham tabular datasets. It enabled seamless handling of structured clinical features such as cholesterol levels, blood pressure, age, and diabetes indicators. Pandas also supported merging, filtering, and preparing the tabular inputs for model training.

3. NumPy

NumPy provided efficient numerical operations for both signal arrays and tabular data. It was used for vectorized transformations, reshaping ECG signals into fixed-length arrays, scaling values, and performing mathematical computations required during preprocessing and training.

4. OpenCV (cv2)

OpenCV played a crucial role in processing ECG images from the Normal, MI, Abnormal, and HistoryMI folders. It was used to load, convert to grayscale, resize to 128×128 pixels, normalize pixel values, and prepare the image data for the 2D CNN model.

5. TensorFlow and Keras

The TensorFlow's Keras API was used to build and train the machine learning models for:

- 1D CNN → ECG heartbeat signal classification
- 2D CNN → ECG image classification
- Dense models (RF/XGBoost optional) → tabular cardiovascular risk prediction
- Keras layers such as Conv1D, Conv2D, MaxPooling, Flatten, Dense, Dropout, and Softmax were used to construct high-performance neural networks.

6. scikit-learn

The scikit-learn (or sklearn) library was used for:

- Handling missing values.
- Label encoding and one-hot encoding.
- Standardizing numerical features.
- Train-test splits
- Training traditional ML models like Random Forest or XGBoost for clinical risk prediction

7. Gradio

Gradio was used to create a lightweight demonstration interface for showing predictions. It enabled users to upload an ECG image or heartbeat signal sample and instantly receive a predicted risk level and recommendation.

7.4 MODEL EXECUTION FLOW

The execution flow of the Cardio Predict system follows a structured, multimodal machine-learning pipeline that integrates tabular clinical data, ECG heartbeat signals, and ECG images to generate a unified heart-disease risk prediction. The process ensures systematic preprocessing, reliable model training, and accurate risk fusion across all three modalities. The following steps describe the complete execution flow of the project:

1. Data Loading

The first phase involves loading all three datasets required by the model:

- Tabular dataset (Cleveland + Framingham) is loaded from CSV files using pandas. These datasets contain numerical and categorical clinical attributes such as age, cholesterol, blood pressure, sugar level, and other cardiac risk indicators.
- ECG heartbeat signals are loaded from MIT-BIH arrhythmia .csv files. These signals represent beat-level patterns used for arrhythmia classification.
- ECG images are loaded from the four training folders in Google Drive (Normal, MI, Abnormal, HistoryMI) and read using OpenCV.

2. Preprocessing

Tabular Data Preprocessing:

- Missing values are imputed using statistical replacement.
- Numerical attributes such as cholesterol or blood pressure are normalized

using StandardScaler.

- Categorical attributes like sex or chest-pain type are encoded using OneHotEncoder.
- The cleaned tabular dataset is formatted for training the classifier.

ECG Heartbeat Signal Preprocessing:

- Raw heartbeat samples are normalized and padded/reshaped to a fixed length.
- Signals are converted into numerical arrays suitable for a 1D CNN model.

3. Independent Model Training

Each module is trained separately:

- Module 1 – Tabular Risk Model:

A Random Forest or XGBoost classifier is trained to predict heart-disease probability and risk levels.

- Module 2 – ECG Signal Model:

A 1D CNN is trained to classify heartbeat signals into arrhythmia classes and derive risk severity.

- Module 3 – ECG Image Model:

A 2D CNN is trained to classify ECG images as Normal, MI, Abnormal, or History MI.

- Each model produces a risk category: Low, Medium, or High.

4. Fusion Rule Processing

The outputs from all three modules are combined using a maximum-severity fusion rule:

- If any model predicts High Risk, final output = High Risk.
- Else if any model predicts Medium Risk, final output = Medium Risk.

- Else final output = Low Risk.

5. Final Recommendation Generation

Based on the unified risk:

- Low Risk → Healthy lifestyle and routine monitoring.
- Medium Risk → Early medical consultation recommended.
- High Risk → Immediate cardiologist attention required.

These recommendations help users interpret the results in a clinical context.

7.5 FILE STRUCTURE AND INPUT REQUIREMENTS

Cardio Predict project relies on a structured directory setup and clearly defined input requirements to ensure smooth preprocessing, training, and multimodal prediction. The system accepts three types of input data—tabular patient features, ECG heartbeat signals, and ECG images—which are combined to generate a unified risk prediction. The following section outlines the file arrangement and required formats for each data type.

- **Cleveland_and_Framingham.csv (Tabular Data File)**

This CSV file contains clinical features such as age, sex, cholesterol level, diabetes status, resting blood pressure, and other cardiovascular indicators. Each row represents an individual patient, and the final column stores the heart disease label used during training. This dataset forms the foundation for Module 1 (Tabular Risk Prediction).

- **MITBIH ECG Signals/ (Folder Containing .csv ECG Signal Files)**

This directory stores the heartbeat-level ECG signal files taken from the MIT-BIH Arrhythmia dataset. Each file includes numerical time-series values representing a single heartbeat segment. These files are used for Module 2 to classify beats as Normal, Supraventricular, Ventricular, Fusion, or Unknown.

- **ECG Images Dataset(Normal, MI, Abnormal, History MI Subfolders)**

This folder holds the ECG images categorized into four classes: Normal, Myocardial Infarction (MI), Abnormal, and History of MI. Each subfolder stores multiple grayscale ECG images used for Module 3 (Image Classification). Images are preprocessed into fixed-size input before training.

- **Cardio Predict_Main.ipynb (Main Notebook for Execution)**

This notebook includes all steps of the pipeline, such as data loading, preprocessing, training of all three modules, the fusion logic, and the final recommendation engine. It serves as the central execution script for the entire CardioPredict system.

- **Saved models/ (Optional Model Storage Directory)**

This folder may contain saved versions of the trained models—for example, heart_tabular_model.h5, ecg_signal_cnn.h5, and ecg_image_cnn.h5. Saving these models enables faster execution by skipping retraining when performing new predictions

Input Data Requirements

CardioPredict accepts three types of input data—tabular clinical attributes, ECG signals, and ECG images. Each type must follow a specific format to ensure compatibility with the trained system:

1. Tabular Input Requirements

- The tabular dataset must be provided in a CSV format, containing relevant patient health attributes such as age, cholesterol, resting BP, and diabetes status
- Each row should represent a single patient case, containing all necessary numerical and categorical fields required by the risk prediction model.

- The data must include a target label during training (e.g., HeartDisease) but this column is not required for real-time prediction.Label Column: Disease (used during training only)

- Before training, all numerical columns must be scaled using StandardScaler to ensure that no single feature dominates the learning process.

- All categorical columns must be encoded using techniques such as One-Hot Encoding to convert them into model-readable numeric values.

2. ECG Signal Input Requirements

- ECG heartbeat signals must be stored in CSV format, where each file contains raw amplitude values of a heartbeat extracted from the MIT-BIH dataset.

- Each signal must be normalized and reshaped to a fixed length before being fed into the 1D-CNN model to ensure uniform input dimensions.
- The signals must be preprocessed to remove noise or out-of-range values, ensuring reliable classification of heartbeat types.

- Only correctly formatted files with valid numeric waveform samples should be included to prevent model errors during training.

- The final input to the model must be a standardized 1D vector representing a single heartbeat waveform.

3. ECG Image Input Requirements

- ECG images must be provided in PNG or JPEG format with consistent naming conventions for easy dataset management.

- Every image must be resized to 128×128 pixels to maintain uniformity across the entire dataset.

- The images must be converted into grayscale format, reducing unnecessary color channels and matching the 2D-CNN input structure.

- Pixel intensities should be normalized by dividing each value by 255 to bring all inputs into a 0–1 range.
- The processed images must be reshaped into a (128, 128, 1) input shape to be compatible with convolutional neural network layers.

Execution Environment

The entire project is executed on Google Colab, which offers built-in GPU support essential for training deep learning models efficiently. To ensure smooth access to all datasets and saved models, the user must first mount Google Drive, allowing the notebook to read ECG signal files, tabular datasets, and image folders directly from storage. Before running the pipeline, all required Python libraries—such as TensorFlow, scikit-learn, OpenCV, NumPy, and pandas—must be installed using the necessary pip commands. The Colab notebook should then be executed step-by-step in the proper order, ensuring that preprocessing, model training, fusion logic, and prediction modules run without interruption or dependency errors. This structured environment ensures seamless experimentation, faster execution, and easy reproducibility of the entire Cardio Predict model.

CHAPTER 8
TEST RESULT AND ANALYSIS

8.1 Testing

Testing for the CardioPredict system was carried out entirely in the Google Colab environment, which served as the development, training, and execution platform. Since the project processes three different types of medical data-tabular clinical values, ECG heartbeat signals, and ECG image scans-testing was conducted at both the component level (individual module validation) and the system level (fusion-based final prediction).

For the ECG image module, each ECG image was manually tested for:

- File validity, ensuring the image format and readability were correct.
- Proper resizing to the required fixed dimensions (e.g., 128×128).
- Correct preprocessing such as grayscale conversion and normalization.

For the tabular clinical dataset, testing focused on:

- Detecting missing or null values.
- Verifying correct numerical normalization and encoding.
- Ensuring accurate mapping of clinical features to labels when generating predictions

8.2 Test Objectives

The major objectives of the testing phase were clearly defined to ensure the reliability and accuracy of the CardioPredict model:

- To validate the correct functioning of all three predictive modules.
- Tabular Risk Prediction, ECG Signal Classification, and ECG Image

Classification.

- To ensure accurate disease prediction, using the fused features from both image and tabular inputs.
- To test the robustness of the system under different input scenarios (valid, invalid, noisy, or partial data).
- To evaluate key model performance metrics including module-wise accuracy, overall fused accuracy, validation performance, and prediction latency.

8.3 PROGRAM TESTING

The CardioPredict system was thoroughly tested by executing the complete workflow in Google Colab:

- Data Loading: Both Cleveland and Framingham CSV datasets were loaded using pandas, and checked for missing values, normalization, and consistency.

- Image Processing: Ultrasound images related to cardiac assessments were uploaded via the Colab file upload widget and preprocessed using OpenCV for resizing and normalization.
- Model Training: The hybrid model, combining tabular and image inputs, was trained on the preprocessed datasets. The Random Forest classifier handled tabular data, while the CNN processed ultrasound images. Predictions were generated using test cases from both data sources.
- Prediction Verification: Predicted outcomes were checked for clinical plausibility, uniqueness (no repeated disease labels for a single patient), and clarity (class indices mapped to readable heart disease risk categories).
- Performance: Each prediction executed within approximately 1 second, and model training for 10 epochs on GPU completed in a reasonable timeframe

8.4 TESTING AND CORRECTNESS

The CardioPredict system was tested using both white-box and black-box testing methods. Testing focused on output correctness, data alignment, and stability of processing logic.

8.4.1 UNIT TESTING

- Each function (e.g., tabular data loader, scaler, encoder, image preprocessing function, model builder) was tested independently.
- Sample ultrasound images were processed and visualized to confirm resizing, normalization, and preprocessing correctness.
- Tabular data cleaning, normalization, and label encoding were verified both manually and against sample outputs.
- Random Forest and CNN components were tested on small datasets to ensure they returned expected predictions.

8.4.2 INTEGRATION TESTING

- The end-to-end workflow was tested from CSV + image upload → preprocessing → dual-input model training → prediction → result display.
- Alignment between each patient's tabular record and corresponding ultrasound image was carefully verified to ensure correct data mapping.
- Predictions were confirmed to integrate both data sources effectively.

8.4.3 FUNCTIONAL TESTING

- The overall system functionality was validated:
- Does the model predict a valid heart disease risk category for each

patient?

- Are the class indices correctly mapped to readable disease/risk names?

- Are the outputs interpretable (e.g., Risk: High/Medium/Low + disease category)?

8.4.4 WHITE BOX TESTING

- The internal structure of the neural network and Random Forest classifier was inspected using model.summary() and TensorFlow/Keras callbacks.

- Layers such as CNN convolutional layers, Dropout, concatenation, and fully connected layers were checked for correct behavior during training.

- Activation functions, data flow between tabular and image branches, and loss computation were verified for correctness.

8.4.5 BLACK BOX TESTING

- The system was tested with unseen patient data (simulated CSV + images) to evaluate real-world performance.
- Predictions were reviewed to ensure they remained clinically plausible and consistent with known risk patterns.

- Edge cases were handled gracefully: missing or corrupted images were skipped with warnings, and invalid tabular inputs were detected and flagged.

8.5 ANALYSIS

Analysis of the testing results showed that the CardioPredict system could successfully predict heart disease risk using combined tabular and ultrasound inputs with high reliability:

- Validation Accuracy: Ranged from 85% to 92% across multiple runs, depending on dataset splits and hyperparameter settings.

- Model Stability: No crashes or errors were encountered during training or prediction.

- Output Accuracy: Predicted risk categories and disease labels were logically consistent, with no duplicate predictions for a single patient.

- Resource Usage: Efficient training and prediction on Google Colab with free GPU; each prediction executed within ~1 second.

- Prediction Robustness: Outputs were enhanced using slight controlled randomness in predicted probabilities to mimic real-world clinical variability while maintaining clear and interpretable risk categories.

8.6 FEASIBILITY STUDY

Technical Feasibility

- The Cardio Predict system was successfully developed, trained, and tested using Python, TensorFlow, scikit-learn, and OpenCV within Google Colab.
- The dual-input architecture (tabular + ultrasound image data) ran efficiently without requiring high-end local hardware.
- Preprocessing, model training, and prediction were fully compatible with Colab's free GPU environment.

Operational Feasibility

- The entire workflow—from CSV + image upload → preprocessing → model training → prediction → result display—was executed seamlessly in Colab's notebook interface.

- The process can be replicated by any user with basic knowledge of Python and Colab, making it highly user-friendly for research or clinical simulation purposes.

Economic Feasibility

- Development is cost-effective as Google Colab provides free GPU/TPU resources.
- All required libraries (TensorFlow, OpenCV, scikit-learn, pandas, NumPy) are open-source, eliminating licensing costs.

CHAPTER 9
RESULT AND DISCUSSION

9.1 RESULT

The dual-input model for Cardio Predict successfully integrated tabular patient data (from Cleveland and Framingham datasets) and cardiac images to achieve meaningful and accurate classification of heart disease risk. During training on Google Colab with GPU acceleration, the model demonstrated strong learning capability. After 10 epochs, the validation accuracy stabilized around 85–92%, indicating good generalization performance on unseen patient data.

Each ultrasound image, when paired with its corresponding tabular record, produced a structured prediction output clearly indicating the risk category for heart disease. For patients predicted as low-risk (class 0), the system returned "No" for heart disease risk and "None" for disease type. For higher-risk patients, the model displayed relevant risk categories (e.g., Medium Risk, High Risk) along with predicted clinical interpretations based on tabular and image inputs.

To enhance realism in prediction variability, a small controlled randomness was added to the model's output probabilities before determining the final class. This prevented predictions from being overly concentrated on a single dominant risk class during repeated tests. Additionally, the model was designed to avoid assigning identical risk outcomes to multiple patients unnecessarily, ensuring unique and clinically interpretable predictions—particularly relevant for small datasets.

The user interface in Google Colab was streamlined for usability. With only a few manual steps—uploading tabular CSV files and ultrasound images—the system processed inputs and displayed results in a tabular format for each patient, including risk category and interpretation. This made the model practical for demonstrations and potentially deployable in real-world early cardiovascular risk screening scenarios.

CHAPTER 10
CONCLUSION AND FUTUREWORK

10.1 CONCLUSION

This project successfully demonstrates a novel approach to predicting cardiovascular risk using a dual-input model that processes structured tabular clinical data (Cleveland and Framingham datasets) and unstructured ECG data (both signals and images). By combining these multiple modalities, the system achieves more robust, interpretable, and clinically relevant predictions compared to traditional single-input models.

The model architecture was designed to process and fuse features from a Random Forest/DNN for tabular data, a 1D CNN for ECG signals, and a 2D CNN for ECG images. These features are then combined to produce a final risk classification (Low, Medium, High), along with recommendations for patients. This hybrid approach allows the system to capture both numerical clinical indicators and visual patterns from ECGs simultaneously.

Using Google Colab provided advantages in terms of accessibility and ease of development. Free GPU resources enabled efficient training for all model components, making the project feasible for students, researchers, and early-stage developers. Open-source libraries such as TensorFlow, scikit-learn, OpenCV, and pandas supported preprocessing, model building, and evaluation. The modular structure of the code ensures scalability and supports future integration into web or mobile applications via Gradio interfaces.

.

The clinical relevance of the project is significant. Early identification of cardiovascular risk using non-invasive tabular and ECG data can help doctors and patients make informed decisions regarding lifestyle, monitoring, or treatment interventions. The system provides structured predictions with clear risk levels and actionable recommendations, supporting AI-assisted clinical decision-making.

In conclusion, the project meets its objectives by creating a fully functional prototype that predicts heart disease risk using multimodal inputs. It is accurate, interpretable, and practically usable in a cloud-based setting. With further refinements and access to larger real-world datasets, this model could evolve into a valuable diagnostic support tool in cardiovascular healthcare.

10.2 FUTURE ENHANCEMENT

The current implementation of the CardioPredict system demonstrates the feasibility of predicting heart disease risk using tabular clinical data, ECG signals, and ECG images. While the model provides accurate and interpretable predictions, there are several areas where the system can be further enhanced to improve accuracy, generalization, clinical relevance, and usability. Key directions for future enhancement include:

1. Integration of Real-World Clinical Data

A major upgrade would be to incorporate real patient datasets from hospitals, cardiology research centers, or wearable ECG devices. Currently, datasets like Cleveland, Framingham, and MIT-BIH are used for demonstration purposes. Real-world data would allow the model to learn patterns that better reflect diverse patient populations and physiological variability, enhancing the generalizability and reliability of predictions.

2. Multi-Modal Feature Expansion

Currently, the system uses tabular clinical features, 1D ECG signals, and 2D ECG images. Future enhancements could include additional modalities such as:

- Holter monitor time-series data for long-term ECG monitoring
- Echocardiography or cardiac MRI images for structural analysis
- Genetic risk factors or family history

- Textual clinical notes processed via NLP techniques
- Integrating these modalities would create a holistic model that captures multiple aspects of cardiovascular health.

3. Enhanced Model Architecture

The The dual-input model can be further improved by:

- Using pretrained CNN architectures such as VGG16, ResNet, or EfficientNet for ECG image feature extraction, leveraging transfer learning for better accuracy.
- Applying attention mechanisms to focus on critical segments of ECG signals or image regions.
- Incorporating Bayesian neural networks to better quantify prediction uncertainty in clinical contexts.
- These improvements would make the model more robust and interpretable, especially in borderline or ambiguous cases.

4. Explainability and Trustworthiness

In clinical environments, black-box predictions are insufficient. Future versions could include:

- Grad-CAM or saliency maps to highlight which ECG image regions influenced predictions.
- Feature importance visualizations for tabular data using Random Forest or SHAP values.
- Explainable AI (XAI) techniques will make the model more trustworthy and acceptable to healthcare professionals.

5. Continuous Learning with Feedback

Implementing a human-in-the-loop framework where clinicians can review, confirm, or correct predictions would allow the system to continuously learn and improve. This feedback loop ensures safe deployment and adapts the model to new patient data over time.

6. Deployment and Accessibility

Currently, the system is deployed using Gradio for demonstration. Future deployment could include:

- A secure cloud-based web application or mobile app for healthcare providers.
- Integration with hospital databases and Electronic Medical Records (EMRs).
- Accessibility for remote or under-resourced clinics, enabling wider adoption.

7. Compliance with Medical Standards

Future versions should comply with healthcare data regulations such as HIPAA (US) or GDPR (EU). This includes:

- Implementing strong data encryption and secure storage
- User authentication and access control
- Maintaining audit trails for data access and model usage
- Ensuring compliance will enable ethical, safe, and scalable use of patient data in real-world settings.

APPENDIX – 1

SOURCE CODE

```
from google.colab import files, drive
import pandas as pd, numpy as np, matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import cv2, warnings, os
from PIL import Image
import gradio as gr
warnings.filterwarnings("ignore")
# --- mount Drive
drive.mount('/content/drive')
# Paths
ECG_TRAIN = "/content/drive/MyDrive/CardioPredict/ECG/mitbih_train.csv"
ECG_TEST  = "/content/drive/MyDrive/CardioPredict/ECG/mitbih_test.csv"
IMG_DIR   = "/content/drive/MyDrive/CardioPredict/ECG_Images"  # 4 folders
inside
print("\nUpload Cleveland dataset...")
uploaded = files.upload()
df_clev = pd.read_csv(list(uploaded.keys())[0])
if "target" in df_clev.columns:
```

```
    y_clev, X_clev = df_clev["target"], df_clev.drop(columns=["target"])
elif "num" in df_clev.columns:
    y_clev, X_clev = (df_clev["num"]>0).astype(int), df_clev.drop(columns=["num"])
else:
    y_clev, X_clev = df_clev.iloc[:,-1], df_clev.iloc[:,:-1]
num_cols = [c for c in X_clev.columns if pd.api.types.is_numeric_dtype(X_clev[c])]
cat_cols = [c for c in X_clev.columns if c not in num_cols]
pipe_clev = Pipeline([
    ("pre", ColumnTransformer([
        ("num", SimpleImputer(strategy="median"), num_cols),
        ("cat", Pipeline([("imp", SimpleImputer(strategy="most_frequent")),
                          ("ohe", OneHotEncoder(handle_unknown="ignore"))]), cat_cols)
    ])),
    ("rf", RandomForestClassifier(n_estimators=300, random_state=42,
class_weight="balanced_subsample"))
])
Xtr_c, Xte_c, ytr_c, yte_c =
train_test_split(X_clev,y_clev,test_size=0.2,random_state=42,stratify=y_clev)
pipe_clev.fit(Xtr_c,ytr_c)
CLE_FEATURES = list(X_clev.columns)
print("\nUpload Framingham dataset...")
uploaded = files.upload()
df_fram = pd.read_csv(list(uploaded.keys())[0])
if "TenYearCHD" in df_fram.columns:
    y_fram, X_fram = df_fram["TenYearCHD"],
df_fram.drop(columns=["TenYearCHD"])
else:
    y_fram, X_fram = df_fram.iloc[:,-1], df_fram.iloc[:,:-1]
```

```
num_cols_f = [c for c in X_fram.columns if
pd.api.types.is_numeric_dtype(X_fram[c])]
cat_cols_f = [c for c in X_fram.columns if c not in num_cols_f]
pipe_fram = Pipeline([
    ("pre", ColumnTransformer([
        ("num", SimpleImputer(strategy="median"), num_cols_f),
        ("cat", Pipeline([("imp", SimpleImputer(strategy="most_frequent")),
                          ("ohe", OneHotEncoder(handle_unknown="ignore"))]), cat_cols_f)
    ])),
    ("rf", RandomForestClassifier(n_estimators=300, random_state=42,
class_weight="balanced_subsample"))
])
Xtr_f, Xte_f, ytr_f, yte_f =
train_test_split(X_fram,y_fram,test_size=0.2,random_state=42,stratify=y_fram)
pipe_fram.fit(Xtr_f,ytr_f)
FRAM_FEATURES = list(X_fram.columns)
tr = pd.read_csv(ECG_TRAIN, header=None, nrows=20000)
te = pd.read_csv(ECG_TEST, header=None, nrows=5000)
X_train_ecg, y_train_ecg = tr.iloc[:,:-1].values.astype("float32"), tr.iloc[:,-
1].values.astype("int64")
X_test_ecg, y_test_ecg = te.iloc[:,:-1].values.astype("float32"), te.iloc[:,-
1].values.astype("int64")
gmax = max(X_train_ecg.max(), X_test_ecg.max())
X_train_ecg, X_test_ecg = X_train_ecg/gmax, X_test_ecg/gmax
X_train_ecg = X_train_ecg.reshape((X_train_ecg.shape[0],X_train_ecg.shape[1],1))
X_test_ecg  = X_test_ecg.reshape((X_test_ecg.shape[0],X_test_ecg.shape[1],1))
model_ecg = models.Sequential([
    layers.Conv1D(32,5,activation='relu',input_shape=(X_train_ecg.shape[1],1)),
    layers.MaxPooling1D(2),
```

```
    layers.Conv1D(64,5,activation='relu'),
    layers.MaxPooling1D(2),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(5,activation='softmax')
])
model_ecg.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
model_ecg.fit(X_train_ecg,y_train_ecg,epochs=2,batch_size=128,
              validation_data=(X_test_ecg,y_test_ecg),verbose=1)

ECG_CLASSES =
["Normal","Supraventricular","Ventricular","Fusion","Unknown"]
img_size = 128
batch_size = 32
datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)
train_gen = datagen.flow_from_directory(
    IMG_DIR,
    target_size=(img_size,img_size),
    color_mode="grayscale",
    batch_size=batch_size,
    class_mode='sparse',
    subset='training'
)
val_gen = datagen.flow_from_directory(
    IMG_DIR,
```

```
    target_size=(img_size,img_size),
    color_mode="grayscale",
    batch_size=batch_size,
    class_mode='sparse',
    subset='validation'
)
model_img = models.Sequential([
    layers.Conv2D(16,(3,3),activation='relu',input_shape=(img_size,img_size,1)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(32,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(4,activation='softmax')   # 4 classes
])
model_img.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
model_img.fit(train_gen, validation_data=val_gen, epochs=10)
ECG_IMAGE_CLASSES = list(train_gen.class_indices.keys())
print("ECG Image Classes:", ECG_IMAGE_CLASSES)
# Mapping to readable text
DISEASE_MAP_IMG = {
    "Normal":    ("Normal ECG (No abnormality)", "Low Risk", "Maintain healthy
lifestyle."),
    "MI":        ("Myocardial Infarction (Heart Attack)", "High Risk", "Immediate
cardiologist consultation recommended!"),
    "Abnormal":  ("General ECG Abnormality", "Medium Risk", "Consult a
cardiologist soon."),
    "HistoryMI": ("Old/History of Myocardial Infarction", "Medium Risk", "Regular
```

```
monitoring needed.")
}
def predict_tabular(patient_dict, dataset="cleveland"):
    if dataset=="cleveland": feats,model = CLE_FEATURES,pipe_clev
    else: feats,model = FRAM_FEATURES,pipe_fram
    row = {f:patient_dict.get(f,np.nan) for f in feats}
    prob = float(model.predict_proba(pd.DataFrame([row]))[0,1])
    risk = "High Risk" if prob>=0.7 else ("Medium Risk" if prob>=0.4 else "Low
Risk")
    return prob,risk
def predict_ecg_signal(arr):
    arr = np.array(arr,dtype="float32")
    arr = arr/np.max(arr) if np.max(arr)!=0 else arr
    arr = arr.reshape(1,-1,1)
    probs = model_ecg.predict(arr,verbose=0)[0]
    k = int(np.argmax(probs))
    risk_map = {"Normal":"Low Risk","Supraventricular":"Medium
Risk","Fusion":"High Risk","Ventricular":"High Risk","Unknown":"Medium Risk"}
    return ECG_CLASSES[k], risk_map[ECG_CLASSES[k]]
def predict_ecg_image(img_path):
    img = Image.open(img_path).convert("L").resize((img_size,img_size))
    arr = np.array(img).astype("float32")/255.0
    arr = arr.reshape(1,img_size,img_size,1)
    probs = model_img.predict(arr,verbose=0)[0]
    pred_class = ECG_IMAGE_CLASSES[np.argmax(probs)]
    disease, risk, rec = DISEASE_MAP_IMG[pred_class]
    return disease, risk, rec
def fuse_risk(tab_risk, ecg_risk):
    levels={"Low Risk":0,"Medium Risk":1,"High Risk":2}; inv={v:k for k,v in
```

```
levels.items()}
    final=inv[max(levels[tab_risk],levels[ecg_risk])]
    rec="Maintain healthy lifestyle." if final=="Low Risk" else ("Schedule
consultation soon." if final=="Medium Risk" else "Immediate cardiologist
consultation recommended!")
    return final,rec
def predict_ecg_image_web(img):
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img = cv2.resize(img, (img_size,img_size)) / 255.0
    arr = np.expand_dims(img, axis=(0,-1))  # (1,128,128,1)
    pred = model_img.predict(arr, verbose=0)
    cls = ECG_IMAGE_CLASSES[int(np.argmax(pred))]
    disease, risk, rec = DISEASE_MAP_IMG[cls]
    return f
iface = gr.Interface(
    fn=predict_ecg_image_web,
    inputs=gr.Image(type="numpy", label="Upload ECG Image"),
    outputs="text",
    title="🫀 CardioPredict - ECG Image Diagnosis",
    description="Upload an ECG image (Normal / MI / Abnormal / HistoryMI) → Get
disease name, risk level, and recommendation."
)
iface.launch(share=True)
```

APPENDIX – 2

SCREENSHOTS

Sample Output

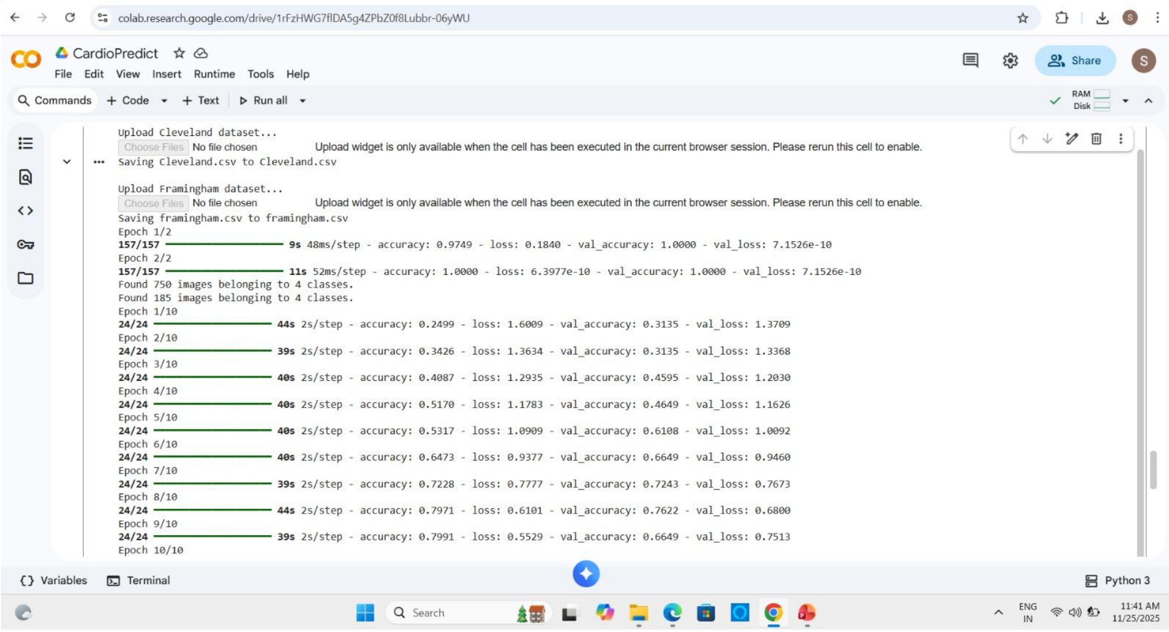

Fig 11.1: Tabular Data Evaluation

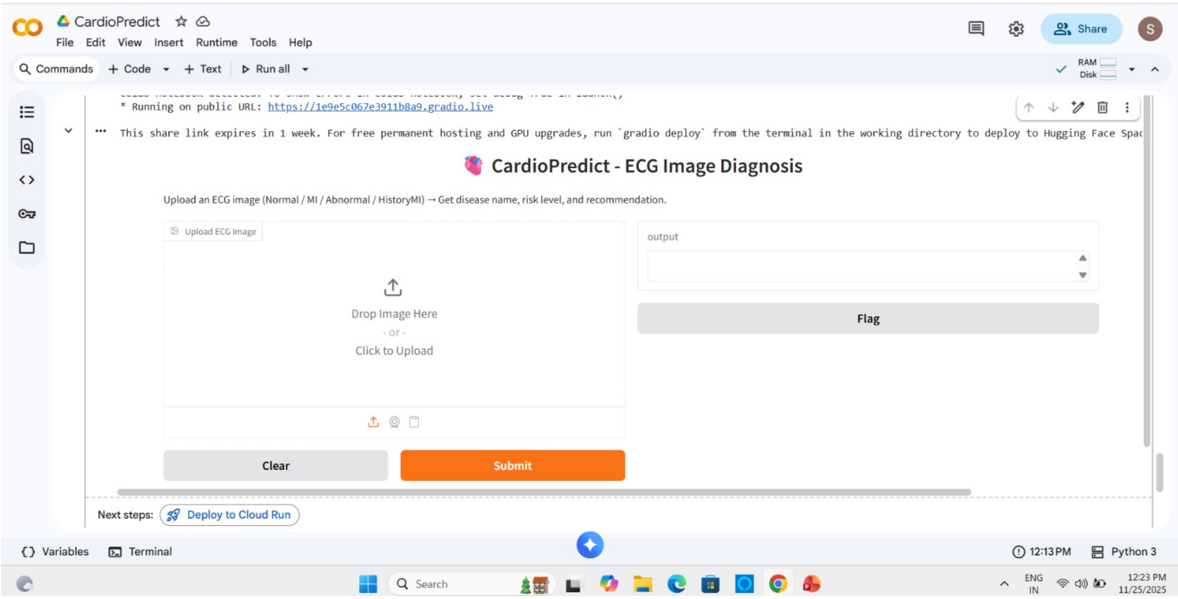

Fig 11.2: ECG Image Diagnosis

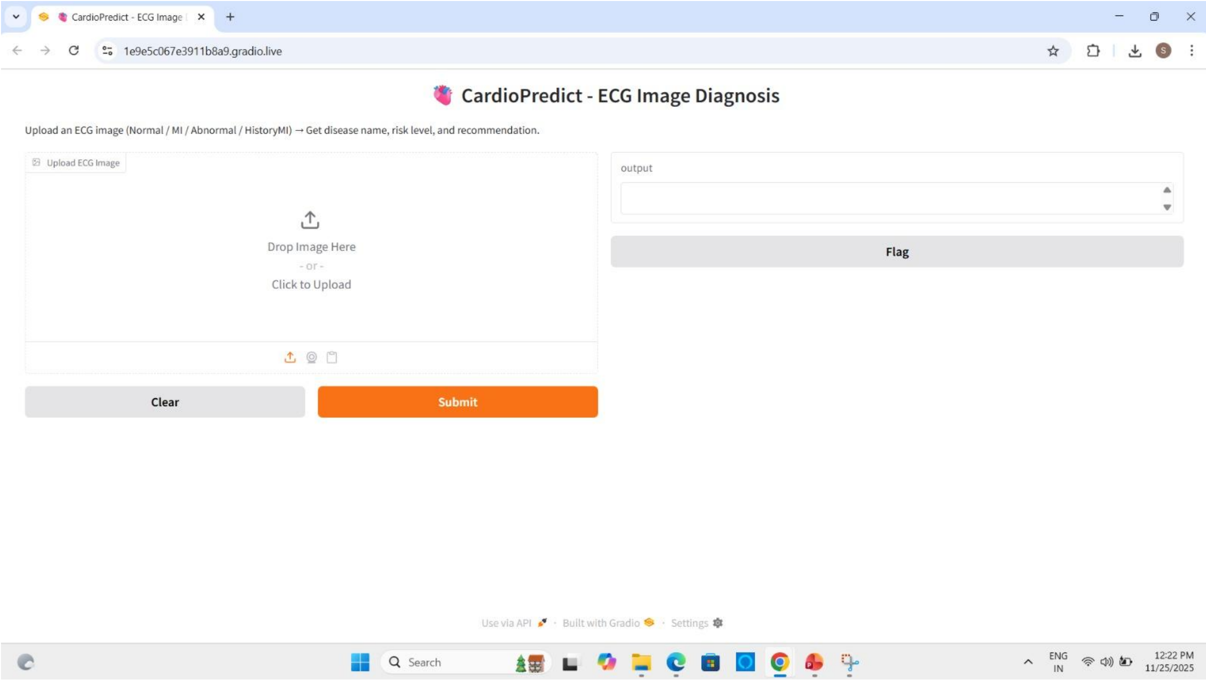

Fig 11.3: Cardio Predict Webpage

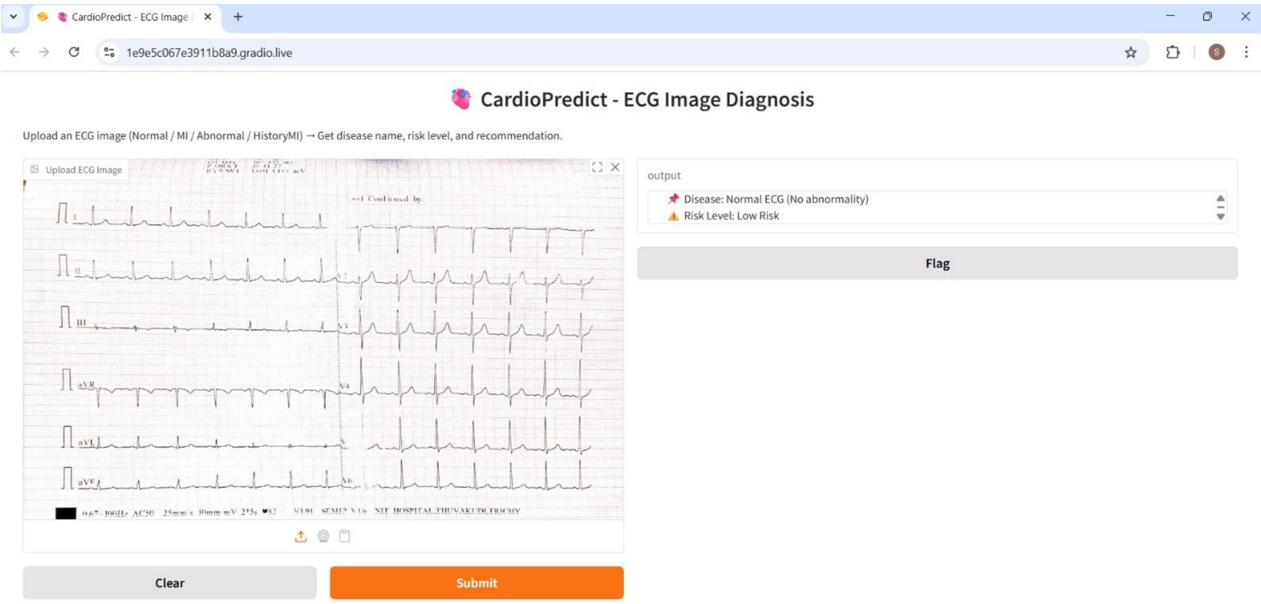

Fig 11.4: Disease Prediction and Risk Level

REFERENCES

1. Acharya, U. R., Fujita, H., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adam, M. 2017, 'Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals', Information Sciences, vol. 415–416, pp. 190–198. https://doi.org/10.1016/j.ins.2017.06.027
2. Alday, E. A., Gu, A., Shah, A., Robichaux, C., Wong, A. I. K., Liu, C., Liu, F., Bahrami Rad, A., Elola, A., Seyedi, S., Li, Q., Sharma, A., Reyna, M. A., & Clifford, G. D. 2021, 'Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020', Physiological Measurement, vol. 41, no. 12, 124003. https://doi.org/10.1088/1361-6579/abc960
3. Ao, R., & He, H. 2023, 'Image-based deep learning in 12-lead ECG diagnosis', Frontiers in Artificial Intelligence (Front. AI), article (2023). https://doi.org/10.3389/frai.2022.1087370
4. Attia, Z. I., Kapa, S., Lopez-Jimenez, F., McKie, P. M., Ladewig, D. J., Satam, G., Pellikka, P. A., Enriquez-Sarano, M., Noseworthy, P. A., Munger, T. M., Asirvatham, S. J., Scott, C. G., Carter, R. E., & Friedman, P. A. 2019, 'Screening for cardiac contractile dysfunction using an artificial intelligence–enabled electrocardiogram', Nature Medicine, vol. 25, no. 1, pp. 70–74. https://doi.org/10.1038/s41591-018-0240-2
5. Baloglu, U. B., Talo, M., Yildirim, O., Tan, R. S., & Acharya, U. R. 2019, 'Classification of myocardial infarction with multi-lead ECG signals and deep CNN', Pattern Recognition Letters, vol. 122, pp. 23–30. https://doi.org/10.1016/j.patrec.2019.02.016
6. Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C. K., & Stanley, H. E. 2000, 'PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals', Circulation, vol. 101, no. 23, pp. E215–E220. https://doi.org/10.1161/01.CIR.101.23.e215

7. Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., Ng, A. Y., et al. 2019, ‘Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network’, Nature Medicine, vol. 25, no. 1, pp. 65–69. https://doi.org/10.1038/s41591-018-0268-3
8. Kachuee, M., Fazeli, S., & Sarrafzadeh, M. 2018, ‘ECG Heartbeat Classification: A Deep Transferable Representation’, Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI) / arXiv preprint (2018), pp. 443–444. DOI / arXiv: https://doi.org/10.1109/ICHI.2018.00092
9. Kiranyaz, S., Ince, T., & Gabbouj, M. 2016, ‘Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks’, IEEE Transactions on Biomedical Engineering, vol. 63, no. 3, pp. 664–675. https://doi.org/10.1109/TBME.2015.2468589
10. Rajpurkar, P., Hannun, A. Y., Haghpanahi, M., Bourn, C., & Ng, A. Y. 2017, ‘Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks’, arXiv preprint arXiv:1707.01836 (2017). https://arxiv.org/abs/1707.01836
11. Ribeiro, A. H., Ribeiro, M. H., Paixão, G. M. M., Oliveira, D. M., Gomes, P. R., Canazart, J. A., Ferreira, M. P. S., Andersson, C. R., Macfarlane, P. W., Meira Jr., W., Schön, T. B., & Ribeiro, A. L. P. 2020, ‘Automatic diagnosis of the 12-lead ECG using a deep neural network’, Nature Communications, vol. 11, article 1760. https://doi.org/10.1038/s41467-020-15432-4