CARDIOLOGIST ASSISTANT

Capstone Project Report

End-Semester Evaluation

Submitted by:

(102017195) TWESHA ARVIND

(102017184) ISHITA KAUNDAL

(102017186) PARV GUPTA

(102017142) SHANTAM ANAND

(102017190) VARCHASVA SINGH

BE Fourth Year, COPC

CPG No: 233

Under the Mentorship of Dr.

Maninder Kaur

Associate Professor

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

December 2023

ABSTRACT

ECG is a commonly used non-invasive medical method to evaluate the electrical activity of the heart. Through the acquisition of cardiac electrical signals and patterns, it is vital to the diagnosis of numerous cardiovascular illnesses (CVDs). This research attempts to improve the precision and effectiveness of CVD diagnosis by creating a sophisticated ECG-based illness detection system through machine learning techniques.

The proposed effort makes use of an extensive dataset of ECG recordings that were gathered from a variety of patient demographics, including both healthy persons and those with various CVDs. The ECG data is cleaned and standardized using preprocessing procedures, guaranteeing high-quality input for further analysis. To extract meaningful characteristics from the raw ECG signals, feature extraction techniques are used, with an emphasis on identifying distinctive patterns linked to particular cardiac diseases. A disease classification model is trained using machine learning. Based on the retrieved attributes, the model learns to distinguish between various heart diseases, including arrhythmias, myocardial infarction, and normality. The use of transfer learning techniques to modify pre-trained models for the particular ECG classification task is also investigated. This can assist address the difficulties caused by the scarcity of medical datasets.

The purpose of the proposed ECG-based illness detection system is to give medical professionals a dependable and effective tool to aid in the early and precise identification of cardiovascular disorders. The project's results could greatly cut down on the amount of time and experience needed to interpret ECGs, allowing for quicker treatments and better patient outcomes.

In conclusion, by using machine learning, this effort takes a big step towards improving the capabilities of ECG-based disease identification. Through the integration of strong data preprocessing, insightful feature extraction, and potent classification models, the system aims to enhance the precision and efficacy of cardiovascular disease diagnosis, hence promoting more efficient healthcare practices

DECLARATION

We hereby declare that the design principles and working prototype model of the project entitled Cardiologist Assistant is an authentic record of our own work carried out in the Computer Science and Engineering Department, TIET, Patiala, under the guidance of Dr. Maninder Kaur during 7th semester (2023).

Date: 19th December 2023

Roll No.	**Name**	**Signature**
102017195	Twesha Arvind	
102017184	Ishita Kaundal	
102017186	Parv Gupta	
102017142	Shantam Anand	
102017190	Varchasva Singh	

Counter Signed By:

Faculty Mentor:

Dr. Maninder Kaur

Associate Professor
CSED,
TIET, Patiala

ACKNOWLEDGEMENT

We would like to express our thanks to our mentor Dr. Maninder Kaur. She has been of great help in our venture, and an indispensable resource of technical knowledge. She is truly an amazing mentor to have.

We are also thankful to Dr. Shalini Batra, Head, Computer Science and Engineering Department, entire faculty and staff of Computer Science and Engineering Department, and also our friends who devoted their valuable time and helped us in all possible ways towards successful completion of this project. We thank all those who have contributed either directly or indirectly towards this project.

Lastly, we would also like to thank our families for their unyielding love and encouragement. They always wanted the best for us and we admire their determination and sacrifice.

Date: 19th December, 2023

Roll No.	**Name**	**Signature**
102017195	Twesha Arvind	
102017184	Ishita Kaundal	
102017186	Parv Gupta	
102017142	Shantam Anand	
102017190	Varchasva Singh	

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CA	Cardiologist Assistant
CVD	Cardiovascular diseases
ECG	Electrocardiography
SVM	Support Vector Machine
RF	Random Forest
ET	Extra Tree
LR	Linear Regression
DT	Decision Tree
AdaBoost	Adaptive Boosting
SMOTE	Synthetic Minority Over-sampling Technique
CNN	Convolutional neural networks
RNN	Recurrent neural networks
AI	Artificial Intelligence
MI	Machine Learning
CAD	Coronary Artery Disease
FrFt	Fourier transform
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
PCA	Principal Component Analysis

1.1 Project Overview

Cardiovascular diseases (CVDs) continue to be a major cause of morbidity and mortality rates, which is a major worldwide health concern. Improving patient outcomes and facilitating effective medical interventions depend heavily on the prompt and precise diagnosis of these disorders. Since electrocardiography (ECG) provides a non-invasive method of recording the electrical activity of the heart, it has long been a mainstay in the diagnosis of cardiac conditions. However, manual ECG trace interpretation is laborious, prone to human error, and susceptible to interobserver variability. The combination of machine learning and ECG analysis has become a viable approach to transforming disease detection in response to these difficulties.

Errors in diagnosis account for the third highest cause of death worldwide. According to a 2013 study published in the Journal of Patient Safety, between 210,000 and 440,000 fatalities occur each year in the US alone as a result of medical errors made during diagnosis. "It's probably one of the, if not the most under-recognized issues in patient safety," said Dr. Peter Pronovost, head of the Armstrong Institute for Patient Safety and Quality at Johns Hopkins, to CNN. We now recognise that a large portion of the harm that was previously considered unavoidable is avoidable." Enhancing the calibre of diagnostic instruments can help reduce these inaccuracies in diagnosis to some degree.

The electrocardiogram (ECG), which shows the electrical impulses in the heart, offers vital information about cardiac health. ECG data are used by clinicians to diagnose anatomical abnormalities, ischemia episodes, and variations in heart rhythm. However, the manual analysis method used in the past might be complex, requiring time and specialised knowledge. These difficulties may be lessened by automated analysis powered by machine learning algorithms, which would enable quick and precise evaluations even in situations with limited resources.

Heart rate can be computed by factoring in the pulse. An ECG is advised if there are any problems with heart disease, heart chamber size, or irregular heartbeat. The following methods can be used to calculate heart rate.

1. To determine heart rate, use the gap between QRS complexes. An electrocardiogram (ECG) measures heart rate and rhythm. It typically consists of large and small squares; large squares are used as a reference, and five little squares are placed in each large square. On the ECG, the tallest portion is the QRS complex. When one QRS complex occurs, it signifies the completion of one heartbeat. Let's take 3.2 as an example, where 3.2 large squares divide up nearby QRS complexes. The heart rate is calculated using the formula below 300/3.2=93.75 is a normal value(As in Table 1).
2. A second approach. On an ECG trace, 30 huge squares correspond to exactly 6 seconds. Multiplying this value by 10, or 6 × 10 = 60 seconds, yields the number of heartbeats per minute, or BPM. If seven beats are recorded in six seconds, for instance, the heart rate is calculated as seven times ten, or 70 beats per minute.

The most frequent irregular heart rhythm, atrial fibrillation, substitutes an unpredictable pattern for the regular heartbeat. Atrioventricular (AV) heart block, bundle branch block, and tachy-brady syndrome are examples of bradycardias, or slow heart rhythms. The rapid heartbeat known as tachycardia includes:

- Ventricular tachycardia (VT)
- Atrial fibrillation (AF)
- Supraventricular tachycardia (SVT)
- Inappropriate sinus tachycardia
- Atrial flutter
- Ventricular fibrillation (VF).

The intricacy of ECG interpretation is ideally matched with machine learning's ability to identify complicated patterns in massive datasets. By means of training algorithms on various sets of ECG records, these models are able to identify minute anomalies that may go unnoticed by the naked eye. The effectiveness of convolutional neural networks (CNNs), recurrent neural networks (RNNs), and ensemble approaches in extracting characteristics from ECG data and accurately predicting disease has been demonstrated. Through the use of transfer learning, these models' potential is further enhanced by utilising knowledge gained from larger datasets.

This project starts the process of creating a cutting-edge machine learning algorithm-powered

ECG-based illness detection system. The goal is to develop an all-encompassing framework that can quickly and precisely diagnose a variety of cardiovascular disorders, from myocardial infarctions to arrhythmias. This study aims to improve diagnostic skills, expedite medical procedures, and eventually transform cardiac care by fusing the accuracy of machine learning with the wealth of information contained inside ECG signals.

We intend to construct an ECG monitor that utilizes an Arduino platform to record the ECG signal.

This would be especially helpful in medical colleges where students can view the ECG signals while the lecturer explains. This will improve the quality of learning tremendously and produce professionals who are more skilled and less likely to make diagnostic mistakes. It also lessens the possibility of human blunders such as misunderstandings. Because it makes use of basic hardware, this device also seeks to be economical.

This project will eliminate human mistake by using artificial intelligence and machine learning to make an intelligent diagnosis. It will also have several benefits, such as telemedicine and self-screening.

TABLE 1: Normal heart rates according to different age groups.

Age	**Normal heart rate(BPM)**
Up to 1 month	70 to 190
From 1 to 11 months	80 to 160
From 1 to 2 Years	80 to 130
From 3 to 4 Years	80 to 120
From 5 to 6 Years	75 to 115
From 7 to 9 Years	70 to 110
10 Years and above	60 to 100

1.1.1 Technical Terminology

1) ECG Signal Recording:

- ECG AD8232 Sensor: Electrocardiogram sensor for recording heart signals.
- Arduino Nano: Microcontroller for processing and transmitting ECG data.
- Electrodes: Components for capturing electrical signals from the patient's body.
- Jumper Wires: Connectors facilitating the linkage of components in the hardware setup.
- USB Cable: Used for data transfer between Arduino Nano and the computer.

2) User Interface:

- WebApp Interface: User-friendly design for interacting with the application effortlessly.
- Model Application Feature: Functionality enabling the application of selected models to ECG images.
- Intuitive Interaction: Easy-to-use interface without the need for extensive user training.

1.1.2 Problem Statement

Disease identification in the existing healthcare system is difficult and expensive, especially in settings with little resources. A highly advanced, portable, and reasonably priced solution that integrates Arduino-based hardware and sensors to enable early illness detection is desperately needed. The goal of this project is to create a portable disease detection device that can quickly and accurately diagnose patients by utilizing machine learning algorithms to analyze pertinent biological and environmental factors. The gadget should be easy to use, accessible to staff without specialized training, and built to be used in a variety of environments, such as point-of-care centers and remote clinics. If this method is put into practice successfully, it could transform the field of illness detection, enhance healthcare outcomes, and help identify and treat serious medical conditions early on.

1.1.3 Goal

The purpose of this project is to develop an Arduino-based medical diagnostic device that specifically targets arrhythmia and myocardial infarction. In order to accurately classify heartbeats, the device will use machine learning algorithms to analyze ECG readings using sensors. The system's goal is to empower users by providing personalized lifestyle suggestions that prioritize stress management, exercise, food, and sleep habits, in addition to disease detection. The interface's ease of use will enable smooth communication, enabling users to provide pertinent data for customized health recommendations. The gadget promises to be portable, accessible, and integrated with healthcare systems. It also aims to inform users about their cardiovascular health and offer real-time diagnostics. The project's ultimate goal is to transform point-of-care health monitoring by encouraging proactive, individualized methods of managing and preventing illness.

1.1.4 Solution

For managing cardiovascular health, our innovative device offers the best possible all-in-one solution. With unmatched accuracy, our small and smart device uses specialized sensors and Arduino technology to analyze ECG impulses. With the use of sophisticated machine learning algorithms, it can differentiate between a myocardial infarction, arrhythmia, and regular heartbeats with ease, offering on-device diagnostics in real time.

Being a one-stop shop, our gadget stands out for having an easy-to-use interface that allows for smooth communication. Users are able to provide lifestyle information, which causes customised suggestions to be generated. Specifically adapted to the identified cardiovascular diseases, this extensive counsel covers nutrition, exercise, stress management, and sleep patterns.

Our device is environment-adaptable, with ports that allow it to be used in both distant clinics and homes. It is made with accessibility and portability in mind. By facilitating smooth data sharing and promoting expert consultation, its integration with healthcare ecosystems makes this possible. Complete protection for user data is guaranteed by strong privacy protocols.

By enabling people to take charge of their cardiovascular health, our gadget essentially redefines point-of-care health monitoring. This gadget is the pinnacle of innovation, offering a comprehensive solution for proactive and customised cardiovascular health management with its unparalleled powers in disease detection and personalised lifestyle coaching.

1.2 Need Analysis

In the current situation, where it is vitally crucial to maintain safety and health and to undertake routine exams on patients who have a history of obesity, heart disease, etc., this initiative will be highly helpful. With the help of this research, patients will be able to self-diagnose without having to be in close contact to their doctors.

Worldwide, there is a deficiency of skilled cardiologists, which makes it difficult to identify patients promptly and accurately. By offering early analysis, an AI assistant can supplement the expertise of skilled cardiologists, increasing patient access to high-quality cardiac treatment. Because it takes so long to analyse ECG data and medical records, traditional cardiac diagnosis can be laborious. This assistant's speedy processing and interpretation of data speeds up diagnosis and enables more timely treatments. In order to reduce the possibility of human mistake while interpreting ECGs and other medical data, AI systems can analyse big datasets with a high degree of consistency and accuracy. This can be a useful teaching tool that helps junior cardiologists and medical students understand difficult concepts and sharpen their diagnostic abilities. In the long run, an AI-based cardiologist assistant may also save healthcare expenses by optimising treatment regimens, cutting down on pointless tests, and expediting diagnosis.

There has been a great deal of study conducted in the "information age" regarding techniques for prompt diagnosis. Forecasting is crucial because it facilitates the management of the likelihood and rate of disease occurrence while also enabling the reinforcement of the actions required to keep the illness under control. The capacity to estimate a person's risk of developing a heart disease is crucial because it helps make informed decisions about lifestyle choices and the benefits and drawbacks of certain behaviours, including smoking, obesity, and alcohol use, depending on their specific risk of developing a heart disease. However, treatment becomes more affordable with early detection, which can lower treatment costs and minimise consequences. The healthcare systems are undergoing a rapid transformation. Scientists have long since developed automated techniques that have simplified and improved risk forecasting. Convolutional neural networks are used to achieve computation efficiency and speed for each patient.

1.3 Research Gaps

1. **"ECG-based Heartbeat Classifiction in Neuromorphic Hardware":**
   A heartbeat categorization method that utilises two lead-ECG signals was proposed by Federico Corradi et al. [1] in 2019 using a recurrent Spiking Neural Network (SNN). A collection of ECG signals with the relevant heartbeat type labelled serves as the training set for the system. By identifying the patterns of spikes produced by the network's neurons, the SNN gains the ability to recognise the various types of heartbeats. Next, a held-out dataset of ECG signals is used to assess the system.

2. **"1D-CADCapsNet: One dimensional deep capsule networks for coronary artery disease detection using ECG signals":**
   A deep learning-based 1D-CapsNet model was proposed by Ertan Butun et al. [2] in 2020 for the automatic identification of coronary artery disease (CAD) using two- and five-second ECG recordings. An encoder and a decoder make up the model. The ECG signal's characteristics are extracted by the encoder, and the decoder categorises the signal as either CAD or normal. A dataset of ECG signals with the accompanying CAD diagnosis labelled is used to train the model. Next, the model's performance is assessed using a held-out dataset of ECG signals. It attains an accuracy of 98.62% for ECG segments lasting five seconds and 99.44% for segments lasting two seconds.

3. **"Empirical Analysis of Machine Learning Algorithms on Imbalance Electrocardiogram Based Arrhythmia Dataset for Heart Disease Detection":**
   In 2021, Shwet Ketu and Pramod Kumar Mishra [3] used a tenfold class validation approach in conjunction with eight machine learning algorithms, including as SVM, KNN, Random forest, and XT, for empirical analysis on ECG-based disease identification. SVM, KNN, RF, ET, Bagging, DT, LR, AdaBoost, and SMOTE were the eight algorithms. Four

criteria were used to assess the eight algorithms' performance: f1-score, accuracy, precision, and recall. The PhysioNet arrhythmia dataset, which includes ECG signals from people with and without heart illness, was the dataset used in the study.

4. **"ECG- Based Machine Learning Algorithm for Heartbeat Classification ":**
   A fusion technique based on FrFT and TERMA was proposed by Saira Aziz, Sajid Ahmed, et al. [4] in 2021 to identify R, P, and T peaks. While traditional wavelet transform techniques were employed to denoise data, the TERMA algorithm's usage of FrFT greatly enhanced peak detection performance. In the MIT-BIH arrhythmia database, they implemented the suggested peak detection technique, and it outperformed the TERMA algorithm by a little margin.

5. **"Detection of P and T-waves in Electrocardiogram":**
   In 2021, T. Vijayakumar, R. Vinothkanna, et al. [5] carried out an examination of ECG signal interpretation using fusion-based feature extraction. There are three steps in the suggested method:

   1. Denoising: A wavelet-based denoising filter is used to reduce the noise in the ECG signal.

   2. Feature extraction: Next, a collection of features, such as time domain, frequency domain, and morphological features, are used to extract the denoised ECG signal.

   3. Fusion: A weighted sum fusion technique is subsequently utilised to fuse the extracted features.

   A dataset of ECG signals from people with and without heart disease is used to assess the suggested approach. The outcomes demonstrated that the suggested approach classified ECG signals with 96.5% accuracy.

1.4 Problem Definition and Scope

Problem Definition:

The work aims at developing a reliable ECG-based detection system, that helps healthcare professionals make timely and accurate diagnoses, leading to better patient outcomes and reduced healthcare costs.

Scope: Here are some potential areas of focus and the scope for such a project

• **Disease Categories**: Patients with arrhythmia, myocardial infarctions (heart attacks), and normality are among the specific cardiac disorders that the study focuses on. Every illness category presents a unique combination of obstacles and possibilities for identification and categorization.

• **Data Collection:** Compile an extensive and varied dataset of ECG recordings, encompassing both healthy and sick instances.

• **User Interface:** Design an intuitive user interface that enables healthcare professionals to upload ECG data and obtain automated diagnostic findings.

• **Clinical Collaboration**: To verify the precision and practical applicability of the model's predictions, work with cardiologists and other medical specialists.

• **Instructional Resources:** Provide instructional materials to help medical professionals and students comprehend how to read and diagnose ECGs.

• **Model Development and Selection**: Select suitable deep learning or machine learning models (CNNs, RNNs, hybrid models) that take into account the time-series and sequential nature of ECG data. Create and hone the models to identify patterns that differentiate across various heart diseases.

1.5 Assumptions and Constraints

Table 2 lists out the Assumptions considered for the project.

TABLE 2:

S.No.	**Assumptions**
1.	The data source is close to real data.
2.	The data set has enough variations to provide the results.
3.	The hardware is sensitive enough to capture live patient data
4.	The ECG data is of good quality and free from noise.
5.	The assistant will be able to follow the cardiologist's instructions and complete tasks accurately and efficiently

Table 3 lists out the Constraints considered for the project.

TABLE 3

S.No.	Constraints
1.	Confined to the given dataset.
2.	Medical practitioners have not validated the results.

1.6 Standards

Adherence to relevant standards is crucial to ensure the system's effectiveness, safety, and accessibility. Here are some standards to consider:

1 **ISO 60601-2-25**: This standard focuses on the performance and safety specifications for electrocardiographs, or ECG recorders.

2 **ISO 11073-10406**: This standard covers the exchange of data, including ECG information, between medical equipment and health information systems.

3 **ISO 12052:** Terms pertaining to ECG equipment are defined in this standard.

4 **ISO 61601-2-47:** This standard addresses the specifications for medical electrical systems and equipment used in cardiovascular disease monitoring and diagnosis.

5 **ISO 80601-2-61**: This standard addresses the fundamental safety and functionality of pulse oximeter apparatus, which may be utilised in conjunction with electrocardiogram (ECG) devices..

1.7 Approved Objectives:

1 Designing an Arduino based ECG monitor using AD8232.

2 Detecting Arrhythmia: An irregular heartbeat is called an arrhythmia. It may result in lightheadedness, chest discomfort, or even a cardiac arrest.

3 Detecting Myocardial infraction: A myocardial infarction (MI), also referred to as a heart attack, happens when blood flow to the heart's coronary artery is reduced or interrupted, harming the heart muscle.

4 Creation of website and deployment of ML model in it.

5 Project also provides user with lifestyle changes to regulate back to normalcy.

6 Provides more accuracy, by lowering the possibility of human mistake in the interpretation

of ECG readings, automated ECG analysis algorithms can contribute to more precise diagnoses.

1.8 Methodology:

1 **Signal acquisition:** Gathering the patient's ECG signals is the initial step in the automated ECG machine's operation.
2 **Signal preprocessing:** The ECG signals must be cleaned up to get rid of any noise or artefacts that can interfere with the signals' ability to be analysed. Amplification, baseline correction, and filtering are a few examples of signal preprocessing methods..
3 **Classification:** Following feature extraction, deep learning algorithms will be employed to categorise the ECG signals into distinct groups according to the existence or absence of specific cardiac diseases.
4 **Interpretation and diagnosis:** After the ECG signals are categorised, the model can interpret the data and diagnose any underlying cardiac problems.

1.9 Project Outcomes and Deliverables:

1 **Accurate Disease Identification:** The model's capacity to correctly identify and categorize diseases based on ECG data is its main achievement. This covers ailments like cardiac infarction, arrhythmias, and normality.
2 **Diagnostic Assistance**: By providing prompt and accurate preliminary diagnoses, the model helps healthcare practitioners make important decisions.
3 **Early Detection**: Because the model is intended to detect illnesses in their early stages, its capacity to recognize ailments before people become critically ill may serve as a barometer for the project's performance.
4 **Reduced False Positives/Negatives:** The model's efficacy may be gauged by how well it reduces false positives, which occur when a disease is mistakenly identified, and false negatives, which occur when a disease is not identified.
5 **Providing lifestyle changes:** Administering lifestyle adjustments associated with the indicated ailment.

1.10 Novelty of Work:

A number of cutting-edge applications in the field of healthcare technology now make use of the potential of ECG-based disease detection to provide early diagnosis and ongoing monitoring. They do, however, only offer a single-lead ECG, which might not be as thorough as a clinical multi-lead ECG and might not catch all possible abnormalities. Furthermore, it should be noted that certain applications may provide false positives or false negatives, underscoring the significance of seeking professional advice in order to ensure a precise diagnosis.

The integration of machine learning models with Arduino-based hardware results in a comprehensive solution that connects physical data collection with cognitive analysis. This integration is a unique method that combines cutting-edge technology with data from the actual world.

Since consumer-facing devices might not pick up on all the subtleties, medical knowledge may be required for the existing devices to interpret results. Healthcare experts use our gadget so they can validate the data and, if necessary, provide additional diagnosis.

Our research tackles the critical requirement for quick insights in medical circumstances by utilizing Arduino to perform real-time ECG analysis. A diverse approach is demonstrated by the use of several machine learning models for disease diagnosis. This variety can improve robustness and accuracy while giving medical professionals a wider view of possible diagnosis.

The interactive GUI makes it easier for patients and medical professionals to communicate with each other. By emphasizing comprehension, this human-centered approach guarantees that patients are knowledgeable of their diseases and treatment options. The convergence of UI design, machine learning, and hardware in our idea is very innovative. It has a thorough comprehension of how technology might change the medical field.

2.1 Literature Survey

The literature survey includes the current knowledge including substantive findings, as well as theoretical and methodology contributions to the topic.

2.1.1 Related Work

- Federico Corradi, et al. [1] in 2019 exploited a recurrent Spiking Neural Network(SNN) to propose a heartbeat classification system that exploits a two lead-ECG signals. The system is trained on a dataset of ECG signals that have been labeled with the corresponding heartbeat type. The SNN learns to identify the different heartbeat types by detecting the patterns of spikes that are generated by the neurons in the network. The system is then evaluated on a held-out dataset of ECG signals. SNN is event-driven, which means that it only consumes power when it is processing spikes. This makes it more energy-efficient than traditional systems, which are always consuming power, even when they are not processing data. Second, the SNN is inherently parallel, which means that it can be implemented on a neuromorphic hardware platform that can exploit the parallelism of the human brain.

- Ertan Butun, et al. [2] in 2020 proposed a deep learning-based 1D-CapsNet model for the automatic detection of CAD(coronary artery disease) from two- and five-second ECG records. The model consists of an encoder and a decoder. The encoder extracts features from the ECG signal, and the decoder classifies the signal as either normal or CAD. The model is trained on a dataset of ECG signals that have been labeled with the corresponding CAD diagnosis. The model is then evaluated on a held-out dataset of ECG signals, and it achieves an accuracy of 99.44% for two-second ECG segments and 98.62% for five-second ECG segments. The model is able to learn hierarchical representations of ECG signals, which makes it more robust to noise and other challenges and it is able to provide a spatial interpretation of the ECG signal, which can be helpful for doctors in understanding the findings.

- Shwet Ketu and Pramod Kumar Mishra [3] in 2021 applied eight machine learning algorithms SVM, KNN, Random forest, XT etc. with a tenfold class validation method for empirical analysis on ECG based disease detection. The 8 algorithms were: SVM, KNN, RF, ET, Bagging, DT, LR, AdaBoost, SMOTE. They evaluated the performance of the eight algorithms using four metrics: accuracy, precision, recall, and f1-score. The dataset used in the study is the PhysioNet arrhythmia dataset, which contains ECG signals from patients with and without heart disease. The dataset is imbalanced, with the majority class (normal ECG signals) outnumbering the minority class (abnormal ECG signals) by a ratio of 10:1. The results of the study showed that the SVM was best followed by the RF. SMOTE was also found to be effective in improving the performance

- Saira Aziz, Sajid Ahmed, et al. [4] in 2021 applied a fusion algorithm based on FrFT and TERMA was proposed to detect R, P, and T peaks. Conventional wavelet transform method were used to denoise signals, whereas the use of FrFT in the TERMA algorithm significantly improved the peak detection performance. They applied the proposed peak detection algorithm in the MIT-BIH arrhythmia database, and it performed slightly better than the TERMA algorithm in the detection of the R peak, while significantly better than it in the detection of the P and T waveforms. After the peak detection, the results were used to and the PR and RT intervals as two features of the ECG signal for the classification.

- T. Vijayakumar, R. Vinothkanna, et al. [5] in 2021 performed fusion based feature extraction analysis of ECG signal interpretation. The proposed method consists of **three steps**:
  - **Denoising**: The ECG signal is denoised using a wavelet-based denoising filter.
  - **Feature extraction:** The denoised ECG signal is then extracted using a set of features, including time domain features, frequency domain features, and morphological features.
  - **Fusion:** The extracted features are then fused using a weighted sum fusion method.

  The proposed method is evaluated on a dataset of ECG signals from patients with and without heart disease. The results showed that the proposed method achieved an accuracy of 96.5% in classifying ECG signals.

TABLE 4: It lists out the Research Findings from the Literature Survey

S.No	**Roll Number**	**Name**	**Paper Title**	**Tools/Technology**	**Findings**	**Citation**
1	102017195	Twesha Arvind	ECG-based Heartbeat Classification in Neuromorphic Hardware	Machine Learning, SNN(Spiking Neural Networks)	The SNN can achieve high accuracy in heartbeat classification, energy-efficient implemented on neuromorphic hardware.	Federico Corradi, et al. [1] in 2019
2	102017184	Ishita Kaundal	1D-CADCapsNet: One dimensional deep capsule networks for coronary artery disease detection using ECG signals	1D-CADCapsNet(deep capsule network)	Provided a spatial interpretation of the ECG signal, automatic detection of CAD from two- and five-second ECG records	Ertan Butun, et al. [2] in 2020
3	102017186	Parv Gupta	Empirical Analysis of Machine Learning Algorithms on Imbalance Electrocardiogram Based Arrhythmia Dataset for Heart Disease Detection	SVM, KNN, RF, ET, DT, LR, AdaBoost, SMOTE	The results of the study showed that the SVM was best followed by the RF. SMOTE was also found to be effective in improving the performance of the algorithms, especially for the minority class.	Shwet Ketu and Pramod Kumar Mishra [3] in 2021
4	102017142	Shantam Anand	ECG-based machine-learning algorithms for heartbeat classification	FrFT and TERMA, peak detection algorithm	Peak detection algorithm performed slightly better than the TERMA	Saira Aziz, Sajid Ahmed, et al. [4] in

						2021
5	102017190	Varchasva Singh	Fusion based Feature Extraction Analysis of ECG Signal Interpretation - A Systematic Approach	Fusion based Feature Selection	The proposed method was able to extract features from both the time domain and frequency domain. , fused features from different domains.	T. Vijayakumar, R. Vinothkanna, et al. [5] in 2021

2.1.2 Research Gaps of Existing Literature

- In his study, Federico Corradi et al. [1], in 2019 investigates the use of neuromorphic hardware for ECG-based heartbeat classification. Broader hurdles in neuromorphic computing include scalability issues unique to analog neuromorphic hardware present a notable research gap, integrating neuromorphic engineering with medicine, emphasising the necessity for solutions to successfully apply conventional neuromorphic techniques in medical settings, and researching modelling capabilities in systems like BrainScaleS-2. These gaps show how much work needs to be done to improve the usefulness of neuromorphic engineering in medical settings, especially for tasks like ECG classification.

- There are some significant research gaps in Ertan Butun's study "1D-CADCapsNet" et al.[2], in 2020, for coronary artery disease (CAD) identification utilising ECG data. The 1D-CADCapsNet model shows encouraging findings, but how well it performs in CAD identification from ECG data is unclear because it hasn't been compared to other deep learning models like CNNs and RNNs. Taking care of this void might offer a more thorough comprehension of the model's effectiveness. Moreover, the work ignores the topic of data augmentation strategies to address overfitting problems, indicating a possible research void in methods to improve the model's generalisation using enriched datasets. While not addressed in the work specifically, the combination of neuromorphic engineering and medicine offers an uncharted path for further investigation in the field. Verifying the model's efficacy in a variety of patient demographics and real-world situations still requires clinical validation. Finally, examining other ECG segment durations outside of the two- and five-second timeframes under investigation may provide information on how to best optimise the model's performance for CAD diagnosis.

- In their study, Shwet Ketu and Pramod Kumar Mishra et al.[3], in 2021, identify areas of possible research need in the field of machine learning for ECG analysis and heart disease identification. Further research is required to address the imbalance in ECG-based arrhythmia datasets; machine learning models for clinical decision-making need to be made more interpretable; deep learning techniques need to be investigated for increased accuracy; algorithm performance needs to be validated in real-world clinical settings; and feature extraction methods should be compared to improve the overall performance of heart disease detection models. These shortcomings provide important opportunities for future research to progress the field and contribute to more reliable and understandable machine learning solutions for heart disease detection, even though they are not specifically addressed in the work.

- Saira Aziz, Sajid Ahmed et al.[4], in 2021, provided possible research directions in their study on ECG-based machine learning for heartbeat classification. To ensure that the suggested algorithm is the best at classifying heartbeats from ECG signals, the comparative analysis of the study should be expanded to include a variety of machine learning algorithms, such as support vector machines and deep learning models. Resolving ECG signal abnormalities is a critical task, hence it is imperative that methods for improved accuracy and dependability be investigated in subsequent studies. A prospective research gap for improving accuracy and broadening generalisation is to investigate the integration of deep learning technologies with ECG analysis. The practical implementation of the proposed algorithm necessitates clinical validation of its real-world application. Additionally, future research could investigate the classification of ECG signals into a wider range of heart rhythm classes, thereby expanding the algorithm's potential utility in clinical settings. These gaps offer important information for developing ECG-based machine learning research, even if they are not mentioned in the publication specifically.

- The study by T. Vijayakumar and R. Vinothkanna et al. [5], in 2021, focuses on the MIT-BIH Arrhythmia Database while discussing P and T-wave detection in electrocardiograms (ECGs). Tested using MITDB, the suggested P-wave detector shows errors in identifying

P waves in AVB III and junctional rhythm ECGs. Additional testing on a variety of ECG datasets with a high level of noise and artefacts is needed in order to improve the evaluation of the detector. Furthermore, a topic of possible research interest that is not covered in this paper is the use of neuromorphic deep spiking neural networks (SNNs) for seizure detection. Future research could benefit greatly from an examination of SNNs in the context of sensitive medical data, with a focus on real-time performance and low power usage.

2.1.3 Detailed problem Analysis

- **Neuromorphic Hardware Constraints**: Unlike standard hardware, neuromorphic hardware imposes unique computational and architectural requirements. These limitations—such as low memory or processing power—may affect how neural networks are developed and put into use for ECG-based illness detection. Optimizing the model's depth and complexity within the constraints of neuromorphic technology is the difficult part of making sure that classification is accurate and efficient. Solving these issues is essential to maximizing the use of Capsule Networks (CapsNets) in practical applications, especially in the medical field.
- **Limited Real-world Applications:** Although CapsNets' design principles show promise, their use in ECG-based disease diagnosis applications is not as common as that of regular neural networks. Determining the actual performance and effectiveness of CapsNets in real-world circumstances is difficult due to the absence of well-established applications and benchmarks. To close this disparity and prove that CapsNets are workable solutions in clinical settings, more investigation and advancement are required.
- **Clinical Validation:** Extensive clinical validation is necessary to determine CapsNets' accuracy and dependability in actual clinical situations. For these models to be successfully incorporated into healthcare practices, it is imperative that they function well across a range of patient demographics and conditions.
- **Hardware and Deployment Constraints**: Hardware limits and computational constraints make it difficult to implement ECG-based illness detection models on devices with limited resources, like wearables. For these models to be widely used in a variety of healthcare settings, effective deployment strategies must be developed.
- **Generalization to Unseen Data:** Models that have been trained on certain datasets may

find it difficult to generalize to populations, conditions, or patient demographics that are not sufficiently represented in the training data. For CapsNets to be useful in a wide range of real-world settings, their generalization skills must be improved.

- **Limited Knowledge Representation:** It's possible that CapsNets won't be able to fully integrate complicated medical knowledge, such as drug interactions and illness mechanisms. Improving the knowledge representation capabilities of the models may help diagnose diseases more accurately and clinically.
- **Limited ECG Leads:** Rather of providing the complete 12-lead ECG, many ECG databases only include a selection of leads, such as lead I, II, or III. In order to increase the robustness and dependability of CapsNets in ECG-based disease diagnosis, techniques to alleviate this constraint must be investigated. This restricted information may have an impact on the model's diagnostic accuracy for specific disorders.Data: Models trained on specific datasets struggle to generalize to different patient

2.1.4 Survey of Tools and Technologies Used

TABLE 5: Showcases the survey of tools and technologies used by our project:

Tools/Technologies	Description
Arduino	It involves programming the breadboard to simulate the heart's electrical activity. Arduino outputs data that mimics ECG signals.
LeNet-5	LeNet-5 is a pioneering convolutional neural network (CNN) developed for handwritten digit recognition, featuring convolutional layers, pooling, and fully connected layers, laying the groundwork for modern CNNs.
AlexNet	AlexNet is a significant CNN model known for its success in the ImageNet Large Scale Visual Recognition Challenge. It introduced deep convolutional neural networks, utilizing multiple convolutional layers and ReLU activation functions.
ResNet-50	ResNet-50 is a deep residual network recognized for its depth. It introduces residual or shortcut connections, allowing for the training of very deep networks, combating the vanishing gradient problem.
VGG-16	VGG-16 is a CNN architecture with 16 layers,

	featuring small-sized convolutional filters. It's known for its simplicity and depth, using multiple 3x3 convolutional layers.
GoogleNet	GoogleNet, also known as Inception v1, emphasizes computational efficiency. It introduces the inception module, incorporating various sizes of convolutional filters within the same layer for improved efficiency and performance
Visual Studio 19	Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

2.1.5 Summary

This project offers a complete strategy to address issues with current ECG-based disease diagnosis methods. Using Arduino for signal gathering offers a flexible and affordable way to gather data. This one-stop shop sets itself apart by precisely diagnosing myocardial infarction, arrhythmia, and regular heartbeat patterns in addition to classifying heart disorders. This refined categorization improves our model's diagnostic accuracy, leading to more precise medical results.

This method is based on the integration of software and hardware. The device achieves real-time detection capabilities by using an advanced machine learning algorithm and an Arduino for signal gathering. This system is unique in that it aims to provide actionable insights in addition to diagnosing cardiac problems. Beyond diagnosis, our system suggests customised lifestyle modifications based on the identified ailment, promoting preventative actions and promoting heart-healthy behaviors.

This concept leverages the versatility of Arduino to overcome the limits associated with traditional hardware, addressing both hardware and deployment constraints. This not only ensures efficient processing and compatibility with neuromorphic hardware constraints, but also makes our system deployable on a wide range of devices, including wearables.

Thorough clinical testing confirms the project's robustness and provides assurance about its practical application. Essentially, our novel method includes signal collection, accurate disease categorization, customised suggestions, and hardware compatibility. With a single resource for lifestyle intervention, continuous health monitoring, and ECG-based disease detection, this comprehensive and user-friendly platform aims to be an all-inclusive solution.

2.2 Software Requirement Specification

2.2.1 Introduction

2.2.1.1 Purpose

The role of the Cardiologist Assistant is to recognize and diagnose a variety of cardiac disorders, such as arrhythmia, myocardial infarction, and normality in patients, by using the distinct electrical patterns of the heart collected in electrocardiogram (ECG) data by hardware based on Arduino. by examining characteristics of the ECG waveform, such as anomalies in the QRS complex, alterations in the ST segment, and inconsistencies in the rhythm. By offering non-invasive insights into the health of the heart, ECG-based illness detection has the potential to improve patient outcomes, guide treatment decisions, and enable early diagnosis, ultimately leading to efficient cardiac care and management.

2.2.1.2 Intended Audience and Reading Suggestions

- **Medical Students and Healthcare Professionals:**
  - **Intended Audience:** A deeper grasp of ECG-based disease identification is sought for by healthcare professionals as well as students studying medicine, cardiology, or related subjects.
  - **Reading Suggestions:** Online courses on sites like Coursera or Medscape, research articles in medical magazines like Circulation or Journal of the American College of Cardiology, and textbooks like Lippincott Williams & Wilkins' "ECG Interpretation Made Incredibly Easy!"

- **Engineers and Technologists:**
  - **Intended Audience:** Engineers, researchers, and developers working on medical device technologies or signal processing related to ECG analysis.
  - **Reading Suggestions:** Research papers on IEEE Xplore related to signal processing for ECG analysis, textbooks like "Bioelectrical Signal Processing in Cardiac and Neurological Applications" by Leif Sörnmo and Pablo Laguna, and conferences like the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

- **Patients and General Public**:
  - **Intended Audience:** Individuals interested in understanding how ECG technology is used for disease detection and monitoring.
  - **Reading Suggestions:** Articles on health websites like WebMD or Mayo Clinic, educational videos on platforms like YouTube, and books like "The Patient's Guide to Heart Valve Surgery" by Adam Pick (includes ECG-related content).

- **Academic Researchers:**
  - **Intended Audience**: Researchers and academics focusing on biomedical engineering, medical technology, or cardiology research.
  - **Reading Suggestions**: Peer-reviewed papers from journals like "Medical & Biological Engineering & Computing," "Computers in Biology and Medicine," and relevant proceedings from conferences like the Computing in Cardiology Conference.

- **Health Tech Entrepreneurs**:
  - **Intended Audience**: Individuals interested in developing startups or businesses related to ECG-based disease detection systems.
  - **Reading Suggestions**: Industry reports from market research firms like MarketResearch.com, articles from business publications like Forbes on healthcare technology trends, and academic research on related technologies.

- **Clinical Practitioners:**
    - **Intended Audience:** Practicing clinicians who want to integrate ECG-based disease detection systems into their patient care.
    - **Reading Suggestions**: Clinical guidelines from professional organizations like the American College of Cardiology (ACC) or the American Heart Association (AHA), articles in medical journals like "Heart" or "Journal of Electrocardiology," and relevant books on cardiology practice.

2.2.1.3 Project Scope

- **Disease Categories**: Disease Categories: Patients with arrhythmia, myocardial infarctions (heart attacks), and normality are among the specific cardiac disorders that the study focuses on. Every illness category presents a unique combination of obstacles and possibilities for identification and categorization.
- **Data Collection**: Compile an extensive and varied dataset of ECG recordings, encompassing both healthy and sick instances.
- **User Interface:** Design an intuitive user interface that enables healthcare professionals to upload ECG data and obtain automated diagnostic findings.
- **Clinical Collaboration:** To verify the precision and practical applicability of the model's predictions, work with cardiologists and other medical specialists.
- **Instructional Resources:** Provide instructional materials to help medical professionals and students comprehend how to read and diagnose ECGs.
- **Model Development and Selection**: Select suitable deep learning or machine learning models (CNNs, RNNs, hybrid models) that consider the time-series and sequential nature of ECG data. Create and hone the models to identify patterns that differentiate across various heart diseases.

2.2.2 Overall Description

2.2.2.1 Product Perspective

Introducing our cutting-edge, Arduino-powered Automated Cardiologist Assistant—a game-changer in the field of cardiac care. This state-of-the-art apparatus produces precise ECG signals with ease, assisting medical practitioners in effective diagnosis and treatment planning. Our helper improves diagnostic accuracy, lowers human error, and expedites patient treatment by using Arduino's precision to simulate real-time cardiac data. Its portable design and user-friendly interface facilitate seamless integration into clinical workflows, encouraging proactive and knowledgeable decision-making. With the help of our Automated Cardiologist Assistant, embrace the future of cardiology and enable medical professionals to deliver cutting-edge, dependable, and patient-centered care.

2.2.2.2 Product Features

This cardiology assistant's features include all of its functions and abilities that help medical professionals use ECG data to diagnose and treat cardiovascular diseases. The following are some essential elements of an automated cardiology assistant:

1. **ECG Data Input:**

   Collects real-time ECG data from Arduino based ECG generator.

2. **Automated Analysis**:

   Employ cutting-edge techniques for machine learning and signal processing to automatically analyze ECG data and spot anomalies or trends that could be signs of cardiac problems.

3. **Multi-Condition Detection:**

   Detect a range of cardiovascular conditions, including arrhythmia, myocardial infraction and normalcy in patients.

4. **Decision Support:**

   Provide evidence-based recommendations for treatment plans and interventions based on the detected conditions on an interactive GUI.

These features collectively contribute to the functionality, usability, and value of the cardiologist assistant, enabling healthcare professionals to make more informed decisions and provide better care for patients with cardiovascular conditions.

2.2.3 External Interface Requirements

The hardware, software, or database system that the project and its many components must interface with is mentioned in the external interface requirements. You may need to take into account the following external interface requirements for this system:

2.2.3.1 User Interfaces

An automated cardiology assistant's user interfaces are essential for enabling data visualization, user interaction, and effective use of the system's capabilities. The following are some instances of user interfaces that may be included in the automated cardiac assistant:

Dashboard Overview:

Providing a comprehensive dashboard that includes summary statistics such as the conditions detected, preventive measures etc.

Visualization Panel:

Displaying visualizations of ECG waveforms to be also seen by healthcare professionals.

By ensuring that healthcare professionals can effortlessly navigate and utilize the features of the automated cardiology assistant, well-designed user interfaces can improve the efficacy and efficiency of their clinical practice.

2.2.3.2 Hardware Interfaces

This Cardiologist Assistant uses Arduino to generate ECG signals of patients. The various components of hardware are:

Electrode sensors are devices that detect electrical signals from the body and send them to the Arduino for processing.

Signal Conditioning Circuit: This circuit produces clear and dependable ECG waveforms by filtering out noise and interference.

The Arduino microcontroller, which forms the basis of the system, is responsible for handling connectivity with other components, processing conditioned signals, and producing the ECG waveform.

Data Output Interface: To enable remote monitoring and storing, the ECG data is output via a variety of interfaces, such as USB, Bluetooth, or wireless networking.

Power Supply: To maintain the system's efficiency, a steady power source, such as batteries or external electricity, is required.

Connectivity Ports: The system's adaptability is increased with interfaces for data transfer, firmware updates, and integration with external devices.

2.2.3.3 Software Interfaces

Several essential elements are integrated into the Cardiologist Assistant software interfaces in order to jointly analyze and interpret the generated ECG signals:

ECG Data Acquisition: Real-time data collecting and monitoring for ECGs is made possible by interfaces, which facilitate smooth interaction with ECG sensors and devices.

Signal processing: To provide accurate and trustworthy analysis, sophisticated signal processing techniques clean and preprocess ECG data.

Automated Analysis: Using ECG data, the system automatically recognizes and categorizes cardiac abnormalities such as arrhythmia, myocardial infarction, and normality using machine learning techniques.

Remote Monitoring: The system can facilitate patient monitoring remotely by sending ECG data and analysis results to medical specialists who are located in different places.

Visualization: To help medical professionals understand the analysis, visual representations of the ECG waveform, identified characteristics, and classification results are provided.

2.2.4 Other Non-functional Requirements

2.2.4.1 Performance Requirements

The following are the performance standards for the position of Cardiologist Assistant (CA):

- **Accuracy:** The CA must be able to produce and analyze medical pictures and data with accuracy, as well as forecast associated cardiac problems such arrhythmia, myocardial infraction, and patient normality.
- **Speed**: To provide patients with prompt diagnosis and treatment, the CA must be able to interpret medical pictures and data rapidly.
- **Scalability:** Given the rise in heart disease patients, the CA should be able to manage a sizable volume of medical data and photos.

Interoperability with other medical devices and systems should be possible for the CA. In order for the CA to smoothly integrate into the healthcare workflow, this is crucial.

2.2.4.2 Safety Requirements

The following safety specifications apply only to our system:

- Once consistent and acceptable accuracy has been attained, all ML models ought to be put into use.
- Users should be made aware of the simulated ECG signals limitations and that they are meant to be used for testing and training purposes only, not for diagnosis.
- The simulated ECG signals should be used and interpreted correctly by users, including students and healthcare professionals.
- The system should be built to be safe and not injure users. It is important to take precautions against physical harm or electrical shocks caused by hardware components.

2.2.4.3 Security Requirements

Here are safety requirements specific to our system:

- Unauthorized users should not be able to access or alter the system due to access controls
- In the event of unexpected behavior or safety concerns, an emergency shutdown mechanism should be accessible to swiftly stop the generation of signals.
- Clearly marked emergency protocols and technical assistance contact details are essential.
- Website should be protected with standard protocols.
- Only valid image format should be accepted from user.

2.3 Cost Analysis

TABLE 6: Cost Analysis for Hardware components:

S.No.	Product Name	Price per unit(INR)	Quantity	Total cost(INR)
1.	Jumper Wires	4/-	5	20/-
2.	Arduino Nano	350/-	1	350/-
3.	AD8232 ECG	600/-	1	600/-
			Total Cost	970/-

2.4 Risk Analysis

- Not every pertinent ECG signal may be produced by the Cardiologist Assistant. Treatment delays or missed diagnosis could result from this.
- There's a chance that the Cardiologist Assistant's ECG signal production is opaque. Cardiologists may find it challenging to comprehend how the CA arrived at its conclusions as a result.
- Unqualified workers may use the Cardiologist Assistant inappropriately. Inaccurate diagnosis or therapy suggestions may result from this.

3.1 Investigative Techniques

TABLE 7: Investigative Techniques used

S.No.	Investigative Projects Techniques	Description of Investigative Techniques	Investigative Projects Examples
1.	Signal acquisition	By attaching the Arduino to an ECG device's electrodes, one can use Arduino to acquire ECG signals. Next, the ECG signal will be read by the Arduino, which will then digitize it. Subsequent processing, displaying, or storing of this data is then possible. The procedures for acquiring an ECG signal with an Arduino board are as follows: 1. Attach the electrodes for the ECG to the Arduino. 2. Create a program that reads an ECG signal and transforms it into digital information. 3. As needed, process, save, or show the ECG data. 1.	It works well for technological research and is appropriate for studies that include signal creation.
2.	Classification	Following feature extraction, deep learning algorithms will be utilized to categorize the ECG signals into groups according to the existence or absence of specific cardiac diseases. Among the most widely utilized algorithms are the following: **1) LeNet-5** is a pioneering convolutional neural network (CNN) developed for handwritten digit recognition, featuring convolutional layers, pooling, and fully connected layers, laying the groundwork for modern CNNs. **2) AlexNet** is a significant CNN model known for its success in the ImageNet Large Scale Visual Recognition Challenge. It introduced deep convolutional neural networks, utilizing multiple convolutional layers and ReLU activation functions.	Classification and segregation-based initiatives. The initiatives primarily leverage databases and information retrieval..

		3) ResNet-50 is a deep residual network recognized for its depth. It introduces residual or shortcut connections, allowing for the training of very deep networks, combating the vanishing gradient problem. **4) VGG-16** is a CNN architecture with 16 layers, featuring small-sized convolutional filters. It's known for its simplicity and depth, using multiple 3x3 convolutional layers. **5) GoogleNet,** also known as Inception v1, emphasizes computational efficiency. It introduces the inception module, incorporating various sizes of convolutional filters within the same layer for improved efficiency and performance	
3.	Interpretation and diagnosis	Following the classification of the ECG signals, the model may diagnose any underlying cardiac problems and offer an interpretation of the ECG data. To do this, the categorized ECG signal is compared to a database of ECG signals from patients with established cardiac diseases. After that, the model may determine which ECG signal characteristics are most suggestive of a specific cardiac ailment. An interactive dashboard may subsequently show the diagnosis' findings. Medical practitioners can use this dashboard to visualize diagnosis results and prescribe treatments.	Utilized in many different domains, such as pattern recognition and spotting issues with machinery and systems

3.2 Proposed Solution

An automated Cardiologist Assistant using Arduino is the suggested solution, which will improve medical education, research, and diagnostic validation. This aide promotes cardiac care and medical education by recreating realistic ECG waveforms. This is a summary of the resolution:

- **ECG Signal Generation**: The system, which is based on Arduino, produces artificial ECG signals that mimic actual heart rhythms. The diversity of ECG waveforms, including normal and pathological patterns such arrhythmias and myocardial infraction and normality in people, are recreated using mathematical models and algorithms.
- **Customizable Scenarios**: The assistant lets users set up situations that mimic various heart rates and cardiac diseases. Because of its adaptability, medical professionals and students can test and practice their diagnostic abilities in a variety of settings.
- **Realistic Electrodes Simulation:** The system can simulate the locations of electrodes and how they affect signal quality. This functionality is crucial for teaching users how to properly position electrodes and comprehend how it affects waveform shape.
- **Training and Education:** The assistant can be used to provide practical instruction for medical students and healthcare professionals. It bridges the gap between theoretical knowledge and actual skill by offering a risk-free environment for effectively diagnosing diseases and interpreting complex ECG patterns.
- **Algorithm Validation:** By using the generated signals, researchers and cardiologists can verify the algorithms used for ECG analysis. This helps to ensure that diagnostic instruments can accurately identify different heart problems.
- **Rapid Prototyping and Research**: Without requiring access to real patient data, researchers may quickly test and prototype new ECG analysis methods using the Arduino-based system. This quickens the creation of cutting-edge cardiac diagnostic instruments.
- **User-Friendly Interface:** The assistant has an easy-to-use interface with heart rate trends, condition labels, and simulated ECG waveforms shown. In addition to receiving immediate feedback on their interpretations, users are able to modify settings and see patterns.
- **Data Logging and Feedback:** The system has the ability to record user interactions, analytic methods, and diagnosis. The system's algorithms can be improved and performance evaluation and learning gaps can be found with the help of this data.
- **Continual Improvement:** Based on user input and new developments in medical knowledge, regular updates may provide new cardiac scenarios, algorithms, and features.

Essentially, the Arduino-powered automated cardiologist assistant provides a cutting-edge approach to medical education, algorithm validation, and research. It contributes to the training of skilled medical professionals, advances the development of diagnostic methods, and eventually improves patient care by producing a variety of realistic-looking ECG signals. The model diagnoses any heart problems that may be present and interprets the ECG data. In order to do this, the categorized ECG signal is compared to a database of ECG signals from patients with established cardiac diseases. Next, the model determines which ECG signal characteristics are most suggestive of a specific heart ailment.

After then, an interactive dashboard shows the diagnosis' findings. Medical practitioners can use this dashboard to visualize diagnosis results and prescribe treatments.

3.3 Work Breakdown Structure

A sort of project management called work breakdown structure gives project deliverables a visual representation. This is achieved by breaking down huge projects into smaller, easier-to-manage groups and displaying important milestones inside a hierarchical structure. This project's work breakdown structure is shown below:

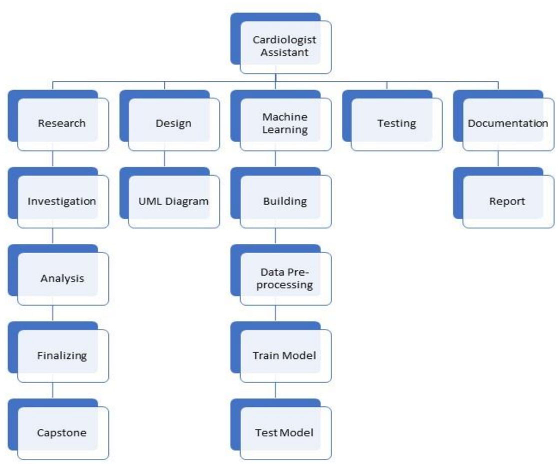

FIGURE 1: Work Breakdown of CA

3.4 Tools and Technology

Tools:

- Arduino Nano
- AD8232 ECG

Algorithms:

- LeNet-5
- AlexNet
- ResNet-50
- VGG-16
- GoogleNet

Software:

- Visual Studio, Google Collab.
- Programming Language: The programming language used for implementing the project is Python.

4.1 System Architecture

A system's architecture reflects its use and its interactions with other systems and the outside world. It explains how every part of the system is connected to every other part and how a data link connects them. A system's architecture is a reflection of how its relationships, functions, and structure are conceptualized. The schematic depicts every element that contributes to the system's operation.

Figure 2 is the block diagram for the “Cardiologist Assistant” project. This block diagram illustrates the essential building blocks of this system and how they collaborate to provide a comprehensive solution for automated disease detection using ECG signals.

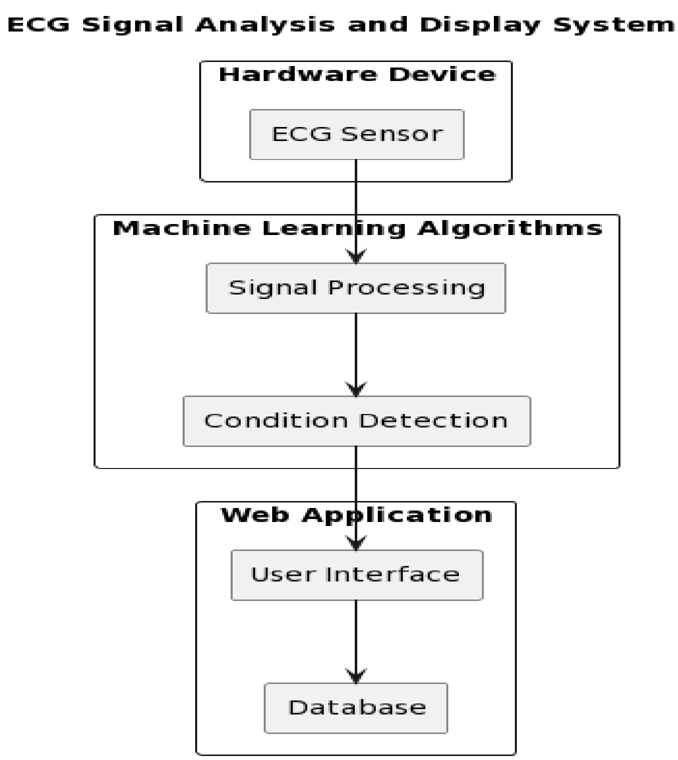

FIGURE 2: Block Diagram

4.2 Design Level Diagrams

4.2.1 Activity / Swimlane Diagram

The workflows for complex procedures, use cases inside use cases, and workflows across use cases are all described in an activity diagram. Activities, states, and transitions between states and activities make up an activity diagram.

This activity diagram shows the sequential flow of tasks in the Cardiologist Assistant system, demonstrating how it assists a medical expert in accurately diagnosing cardiac illness and subsequently offers the necessary lifestyle improvements to return back to normalcy.

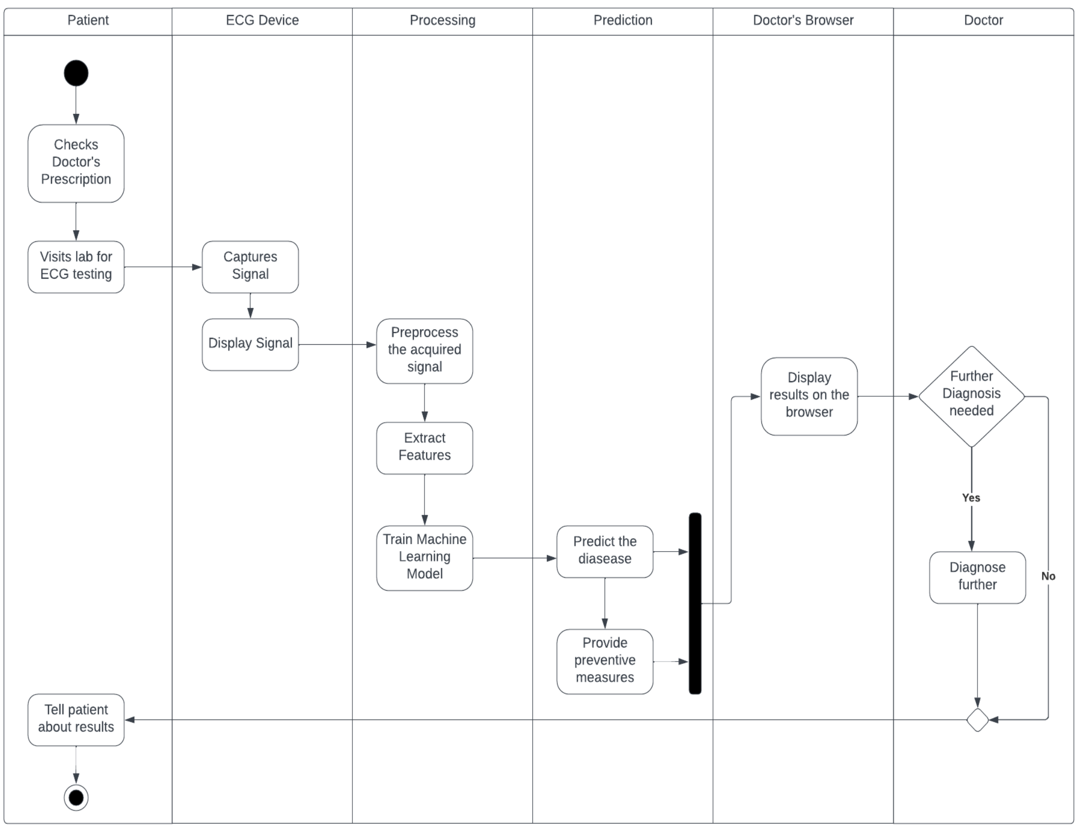

FIGURE 3: Activity/Swimlane Diagram

4.2.1 Sequence Diagram:

A sequence diagram shows how things interact with one another in a sequential fashion. Sequence diagrams show how a system's objects work and in what order.

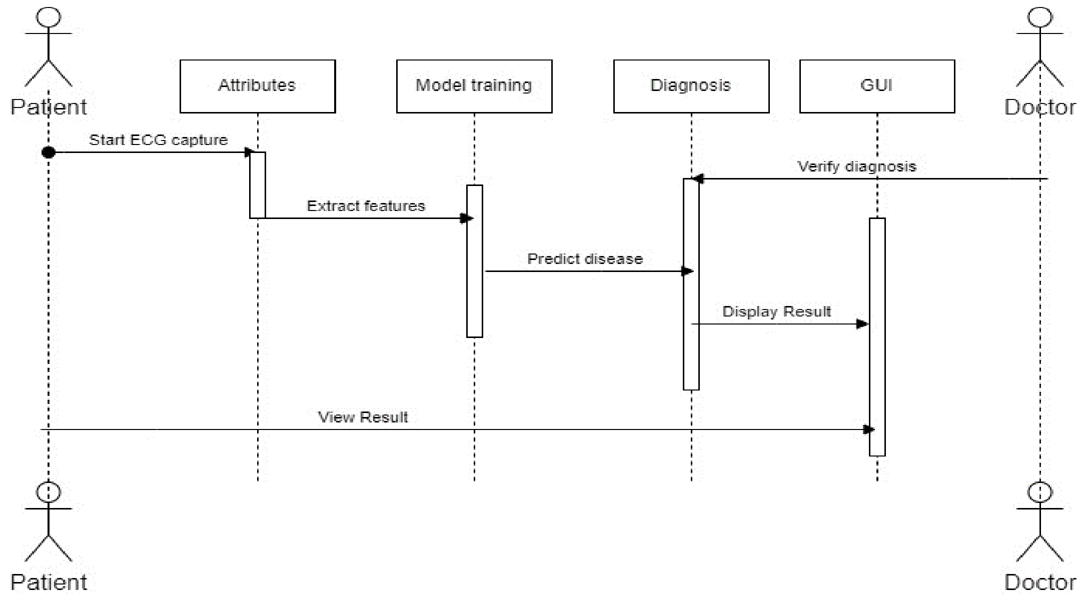

FIGURE 4: Sequence Diagram

4.2.2 ER Diagram

An entity-relationship diagram (ERD) is a graphical representation of a database.

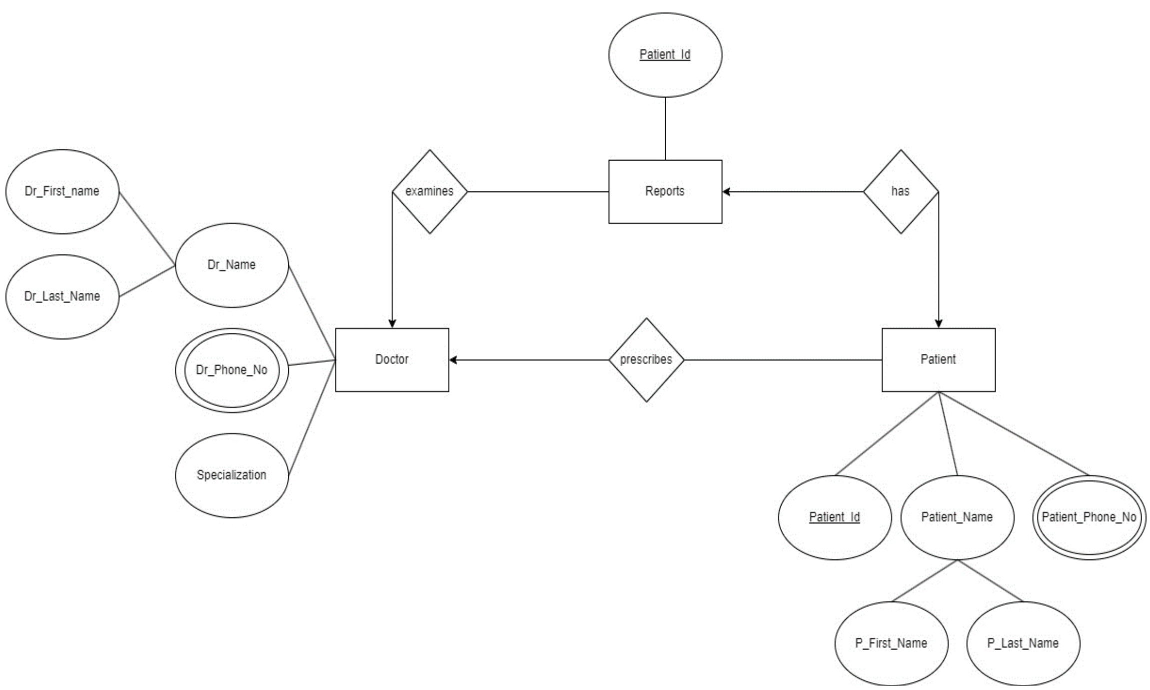

FIGURE 5: ER Diagram

4.2.3 Component Diagram

A UML diagram that depicts a software system's structure is called a component diagram. It illustrates the system's elements, interdependencies, and connections between them.

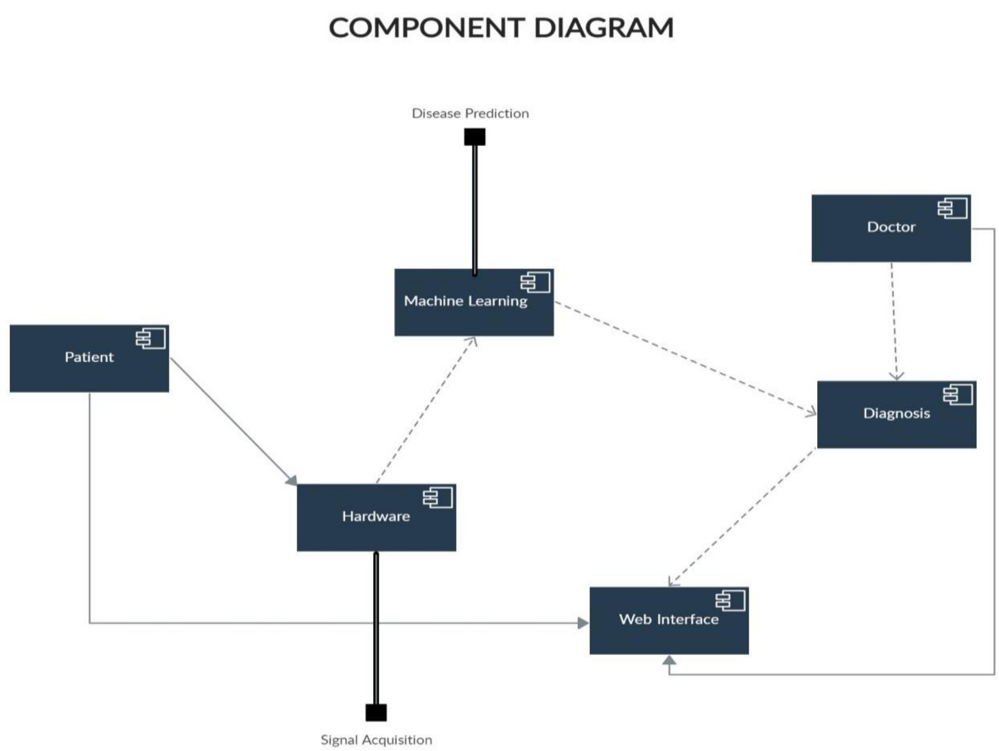

FIGURE 6:Component Diagram

4.2.4 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through a system.

Level 1:

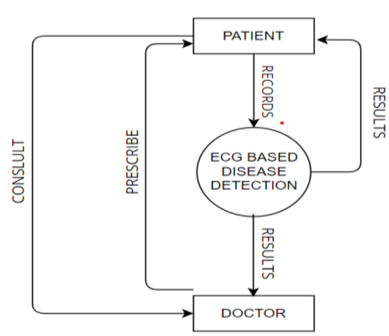

FIGURE 7:Level-1 DFD

Level 2:

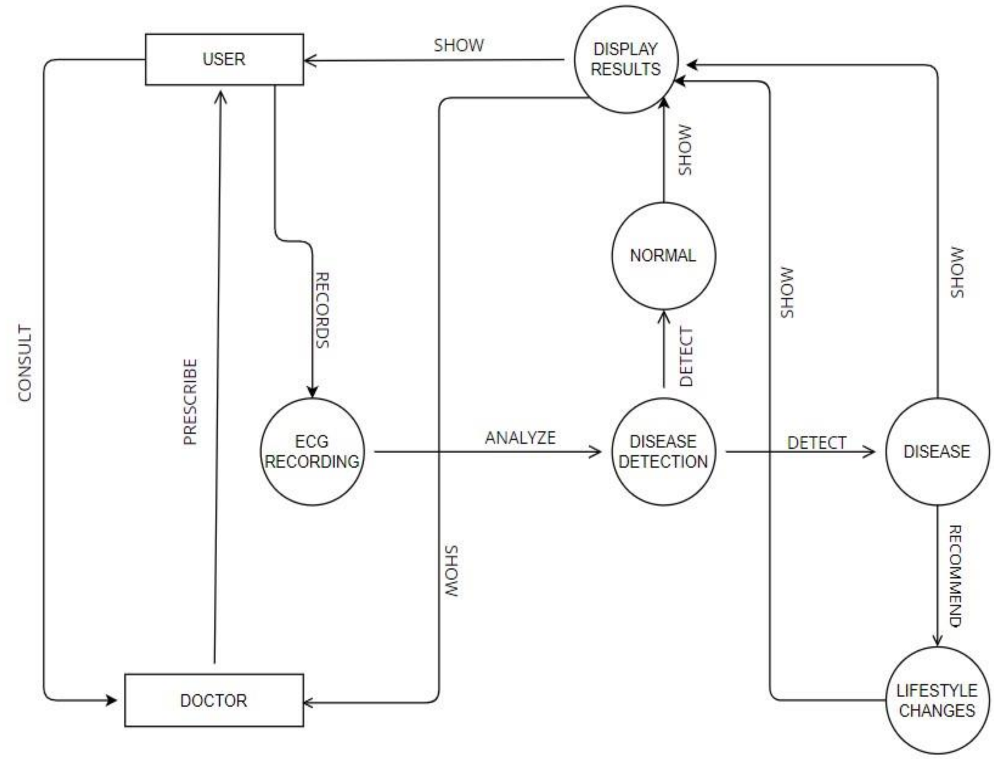

FIGURE 8:Level-2 DFD

4.3 User Interface Diagrams

4.3.1 Class Diagram

In the Unified Modelling Language (UML), a class diagram is a kind of static structure diagram that shows the organization and connections between classes in a system. It gives the classes, their properties, their methods, and the relationships between them a visual representation.

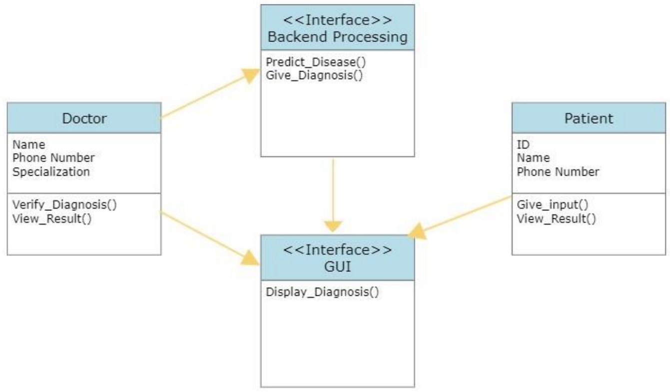

FIGURE 9:Class Diagram

4.3.2 Use Case Diagram

In the Unified Modelling Language (UML), a use case diagram is a kind of behavioral diagram that shows how users (actors) interact with a system to accomplish particular objectives. It offers a high-level overview of the features that a system needs to have for people to engage with it.

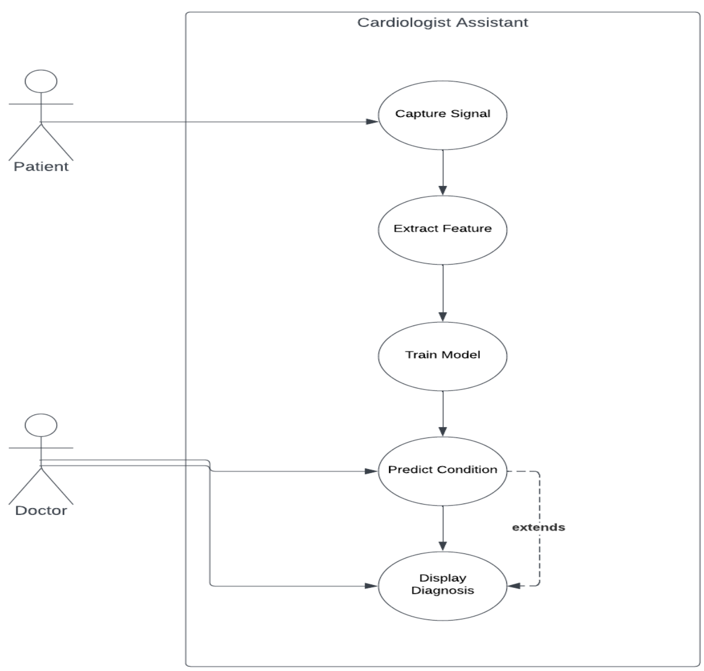

FIGURE 10:Use Case Diagram

4.3.3 Use Case Template

1. **Use Case Title**: Cardiologist Assistant
2. **Abbreviated Title**: CA
3. **Actors:** Patient, Doctor
4. **Description:** In this use case, a hardware device is used to record the patient's ECG readings. Machine learning techniques are then used to analyze the data and identify any heart-related issues. The results and recommended preventive actions are then shown on a website.

   4.1 Pre-Conditions:

   - Patient is connected to the ECG hardware device
   - The hardware device is properly configured and operational
   - Machine learning algorithms and web application are deployed and functional

   4.2 Task Sequence:

   - The ECG hardware device captures the ECG signals from the patient
   - The captured signals are pre-processed and fed into the machine learning algorithms
   - The machine learning algorithms analyze the signals and detect any heart related conditions.
   - The results are sent to the web application.
   - The web application displays the results to the patient and recommends preventive measures

   4.3 Post conditions:

   - The ECG signals are analyzed and abnormal conditions are detected
   - The results are displayed to the patient on the web application along with preventive measures

5. **Authors:** Ishita Kaundal, Twesha Arvind, Parv Gupta, Shantam Anand, Varchasva Singh.

IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1 Experimental Setup

The Cardiologist Assistant project aimed to facilitate disease classification from electrocardiogram (ECG) signal images. There are three main components of our application:

- **Dataset and Data Collection**
- **Machine Learning Models**
- **Hardware Setup**
- **Model Deployment Tools**

Dataset and Data Collection

The dataset utilized for this purpose was obtained from Data Mendeley, comprising a collection of ECG signal images.

Machine Learning Models

We trained five different machine learning models for disease classification based on ECG signal images. The models we used are LeNet-5, AlexNet, VGG 16, GoogleNet, and ResNet 50

Hardware Setup

We built a setup with an Arduino Nano and an ECG AD8232 module, connected by jumper wires, to capture ECG signals. These signals were used as inputs for training our machine learning models to classify diseases based on the ECG images

Model Deployment Tools

Upon successful training, the models were deployed using Streamlit, providing an interactive and user-friendly interface for utilizing the trained models

Tools Utilized:

Development and Training: Google Colab, VS Code

Hardware Integration: Arduino IDE

5.2 Experimental Analysis

5.2.1 Data

This dataset consists of ECG images of patients with cardiac conditions, including both COVID-19 positive and negative cases. It aims to facilitate research on COVID-19 and cardiovascular diseases.

- **Data Files**:
  - ECG Images of Myocardial Infarction Patients (77 images)
  - ECG Images of Patient that have abnormal heart beats (548 images)
  - ECG Images of Patient that have History of MI (203 images)
  - Normal Person ECG Images (859 images)
- **Data Format**:
  - Images are in PNG format.
  - Each image is labeled with the patient's condition.
- **Data Characteristics:**
  - Number of patients: Not explicitly stated, but likely exceeds the total number of images (2,500+).
  - Lead configuration: 12-lead is assumed, as this is the standard for clinical ECG recordings.
  - Signal frequency and duration: Not specified.
  - Image artifacts: May be present, as the dataset aims to be representative of real-world clinical scenarios.
- **Potential Applications**:
  - Development of AI-assisted diagnostic tools for early detection and management of heart conditions.
  - The data is diverse and includes both normal and pathological cases.
  - The dataset is freely available for non-commercial research purposes.

5.2.1 Performance Parameters

A crucial field of medicine that has the potential to save lives by facilitating early diagnosis and treatment is ECG-based cardiac disease detection. An ECG-based system's ability to detect heart illness can be assessed using a number of important performance metrics, including:

Metrics for Classification:

- **Recall:** This quantifies the percentage of real heart disease instances that the technology accurately detects. The system's ability to identify genuine positives is demonstrated by a high sensitivity.

- **Precision** is a metric that quantifies the percentage of correctly predicted favorable outcomes. A high precision means that the system's forecasts of heart disease are trustworthy.

- **F1 Score**: This balanced indicator of categorization performance is derived from the harmonic mean of recall and precision.

An ECG-based cardiac disease detection system's performance evaluation is essential for determining how effective it is and how it might affect clinical practice. Researchers and developers can pinpoint areas for improvement and ultimately produce precise and dependable tools for early diagnosis and management of heart disease by employing an extensive set of performance parameters.

5.3 Working of the Project

5.3.1 Procedural Flow

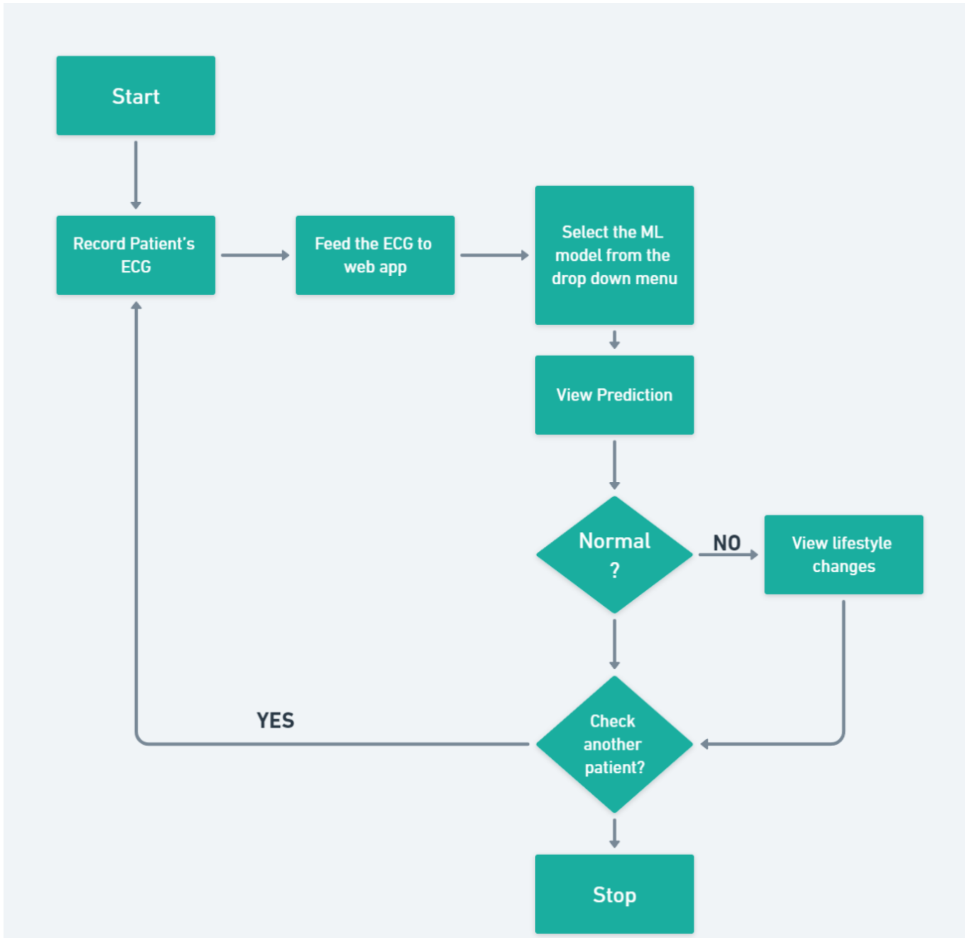

FIGURE 11: Procedural Workflow

The procedural workflow of our application is designed to guide users through a series of sequential tasks aimed at achieving specific financial goals. Each step in the workflow involves three key parameters: input, transformation, and output.

- **ECG Signal Recording and Image Processing**

  1) Input: Hardware records ECG signals from patients.

  2) Transformation: Image processing of ECG signal data into interpretable images.

  3) Output: Processed ECG signal images ready for disease prediction.

- **Disease Prediction and Lifestyle Recommendations**

1) Input: Processed ECG signal images uploaded to the web application.

2) Transformation: Utilizing five different machine learning models for disease prediction and lifestyle recommendations .

3) Output: Predicted disease class and lifestyle recommendations provided for each input image.

- **Interactive Model Selection in Web Application**

1) Input: Image is selected and users choose a specific model among the five available models.

2) Transformation: The selected model processes the uploaded ECG signal image.

3) Output: Display of disease prediction and lifestyle recommendations based on the chosen model.

5.3.2 Algorithmic Approaches

LeNet 5

Here, we have trained LeNet5 architecture on our ECG dataset for 10 epochs .

LeNet 5

```
# Define LeNet-5 model
lenet = models.Sequential()
lenet.add(layers.Conv2D(6, (5, 5), activation='relu', input_shape=input_shape))
lenet.add(layers.MaxPooling2D((2, 2)))
lenet.add(layers.Conv2D(16, (5, 5), activation='relu'))
lenet.add(layers.MaxPooling2D((2, 2)))
lenet.add(layers.Flatten())
lenet.add(layers.Dense(120, activation='relu'))
lenet.add(layers.Dense(84, activation='relu'))
lenet.add(layers.Dense(num_classes, activation='softmax'))  # Adjust num_classes based on your dataset

# Compile the model
lenet.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
lenet.summary()
```

FIGURE 12: LeNet5 architecture

```python
# Train the model
epochs = 10

history = lenet.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator
)

# Save the trained model
lenet_path = f'{model_save_dir}/lenet5_ecg_model.hd5'
lenet.save(lenet_path)

# lenet_loaded_model = tf.keras.saving.load_model(f'{model_save_dir}/lenet5_ecg_model.keras')
```

FIGURE 12: LeNet5 architecture

Alex Net

Here, we have trained Alex Net architecture on our ECG dataset for 10 epochs .

Alex Net

```python
def alexnet_model(input_shape=(64, 64, 12), num_classes=4):
    model = models.Sequential()

    # Layer 1
    model.add(layers.Conv2D(96, (11, 11), strides=(4, 4), activation='relu', input_shape=input_shape))
    model.add(layers.MaxPooling2D((3, 3), strides=(2, 2)))

    # Layer 2
    model.add(layers.Conv2D(256, (5, 5), padding='same', activation='relu'))
    model.add(layers.MaxPooling2D((3, 3), strides=(2, 2)))

    # Layer 3
    model.add(layers.Conv2D(384, (3, 3), padding='same', activation='relu'))

    # Layer 4
    model.add(layers.Conv2D(384, (3, 3), padding='same', activation='relu'))

    # Layer 5
    model.add(layers.Conv2D(256, (3, 3), padding='same', activation='relu'))
    model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

    # Global Average Pooling
    model.add(layers.GlobalAveragePooling2D())

    # Dense layers
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output layer
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

FIGURE 13: Alex Net architecture

```python
# Create the AlexNet model
alex_net = alexnet_model(input_shape=input_shape, num_classes=num_classes)


# Compile the model with a specified learning rate
learning_rate = 0.001  # Set your desired learning rate
optimizer = Adam(learning_rate=learning_rate)
alex_net.compile(optimizer=optimizer,
                 loss='categorical_crossentropy',
                 metrics=['accuracy'])


alex_net.summary()
```

FIGURE 13: Alex Net architecture

```python
# Train the model
epochs = 10

history = alex_net.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator
)

# Save the trained model
alex_net_path = f'{model_save_dir}/alex_net_ecg_model.hd5'
alex_net.save(alex_net_path)
```

FIGURE 13: Alex Net architecture

VGG-16

Here, we have trained VGG-16 architecture on our ECG dataset for 10 epochs .

VGG-16

```python
def vgg16_model(input_shape=(64, 64, 12), num_classes=4):
    model = models.Sequential()

    # Block 1
    model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=input_shape))
    model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 2
    model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'))
    model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

    # Block 3
    model.add(layers.Conv2D(96, (3, 3), activation='relu', padding='same'))
    model.add(layers.Conv2D(96, (3, 3), activation='relu', padding='same'))
    model.add(layers.Conv2D(96, (3, 3), activation='relu', padding='same'))
    model.add(layers.MaxPooling2D((2, 2), strides=(2, 2)))

    # Flatten the output and add Dense layers
    model.add(layers.Flatten())
    model.add(layers.Dense(256, activation='relu'))
    # model.add(layers.Dropout(0.5))
    # model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    # Output layer
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model
```

FIGURE 14: VGG-16 architecture

```python
# Create the VGG16-like model
vgg16_net = vgg16_model(input_shape=input_shape, num_classes=num_classes)

# Compile the model
vgg16_net.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

# Print a summary of the model architecture
vgg16_net.summary()
```

FIGURE 14: VGG-16 architecture

```
# Train the model
epochs = 10

history = vgg16_net.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator
)

# Save the trained model
vgg16_net_path = f'{model_save_dir}/vgg16_ecg_model.hd5'
vgg16_net.save(vgg16_net_path)
```

FIGURE 14: VGG-16 architecture

GoogleNet

Here, we have trained GoogleNet architecture on our ECG dataset for 10 epochs .

GoogleNet

```
def inception_module(x, filters):
    # 1x1 Convolution
    conv1x1_1 = layers.Conv2D(filters[0], (1, 1), padding='same', activation='relu')(x)

    # 1x1 Convolution followed by 3x3 Convolution
    conv1x1_2 = layers.Conv2D(filters[1], (1, 1), padding='same', activation='relu')(x)
    conv3x3 = layers.Conv2D(filters[2], (3, 3), padding='same', activation='relu')(conv1x1_2)

    # 1x1 Convolution followed by 5x5 Convolution
    conv1x1_3 = layers.Conv2D(filters[3], (1, 1), padding='same', activation='relu')(x)
    conv5x5 = layers.Conv2D(filters[4], (5, 5), padding='same', activation='relu')(conv1x1_3)

    # 3x3 MaxPooling followed by 1x1 Convolution
    maxpool = layers.MaxPooling2D((3, 3), strides=(1, 1), padding='same')(x)
    conv1x1_4 = layers.Conv2D(filters[5], (1, 1), padding='same', activation='relu')(maxpool)

    # Concatenate the output of all branches
    inception = layers.concatenate([conv1x1_1, conv3x3, conv5x5, conv1x1_4], axis=-1)

    return inception
```

FIGURE 15: Google Net architecture

```
def googlenet_model(input_shape=(64, 64, 12), num_classes=4):
    input_layer = layers.Input(shape=input_shape)

    # Initial Convolutional Layer
    x = layers.Conv2D(64, (7, 7), strides=(2, 2), padding='same', activation='relu')(input_layer)
    x = layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)

    # Inception Module 1
    x = inception_module(x, filters=[8, 32, 32, 8, 8, 8])

    # Inception Module 2
    x = inception_module(x, filters=[8, 32, 32, 8, 8, 8])

    # Flatten and Dense Layers
    x = layers.Flatten()(x)
    x = layers.Dense(128, activation='relu')(x)
    x = layers.Dropout(0.5)(x)

    # Output layer
    output_layer = layers.Dense(num_classes, activation='softmax')(x)

    model = models.Model(inputs=input_layer, outputs=output_layer)

    return model

# Create the GoogLeNet-like model
googlenet = googlenet_model(input_shape=input_shape, num_classes=num_classes)

# Compile the model
googlenet.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

# Print a summary of the model architecture
googlenet.summary()
```

FIGURE 15: Google Net architecture

```
# Train the model
epochs = 10

history = googlenet.fit(
                        train_generator,
                        epochs=epochs,
                        validation_data=validation_generator
                      )

# Save the trained model
googlenet_path = f'{model_save_dir}/googlenet_ecg_model.hd5'
googlenet.save(googlenet_path)
```

FIGURE 15: Google Net architecture

ResNet50

Here , we have trained ResNet50 architecture on our ECG dataset for 10 epochs.

ResNet50

```python
from tensorflow.keras.applications import ResNet50

# Load the pre-trained ResNet50 model with weights from ImageNet
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=input_shape)

# Freeze the convolutional layers
for layer in base_model.layers:
    layer.trainable = False

# Create a custom head for your specific task
x = base_model.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.5)(x)
output_layer = layers.Dense(num_classes, activation='softmax')(x)

# Combine the base model with the custom head
resnet = models.Model(inputs=base_model.input, outputs=output_layer)

# Compile the model
resnet.compile(optimizer='adam',
               loss='categorical_crossentropy',
               metrics=['accuracy'])

resnet.summary()
```

FIGURE 16: ResNet50 architecture

```python
# Train the model
epochs = 10
history = resnet.fit(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator
)

# Save the trained model
resnet_path = f'{model_save_dir}/resnet_ecg_model.hd5'

resnet.save(resnet_path)
```

FIGURE 16: ResNet50 architecture

Here , we have evaluated our models for their predictions.

Models performance comparison

```python
lenet_path = f'{model_save_dir}/lenet5_ecg_model.hd5'
alex_net_path = f'{model_save_dir}/alex_net_ecg_model.hd5'
vgg16_net_path = f'{model_save_dir}/vgg16_ecg_model.hd5'
googlenet_path = f'{model_save_dir}/googlenet_ecg_model.hd5'
resnet_path = f'{model_save_dir}/resnet_ecg_model.hd5'


#Loading Models:
lenet = tf.keras.saving.load_model(lenet_path)
alex_net = tf.keras.saving.load_model(alex_net_path)
vgg16_net = tf.keras.saving.load_model(vgg16_net_path)
googlenet = tf.keras.saving.load_model(googlenet_path)
resnet = tf.keras.saving.load_model(resnet_path)
```

✓ 26.3s

```python
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

# Function to print evaluation metrics and plot confusion matrix heatmap
def print_evaluation(model, generator, model_name):
    print(f"=================================== Evaluation for {model_name} ===================================")
    y_true = generator.classes
    class_labels = list(generator.class_indices.keys())

    # Get predictions from the model
    y_pred_prob = model.predict(generator)
    y_pred = np.argmax(y_pred_prob, axis=1)


    print("Classification Report:")
    print(classification_report(y_true, y_pred, target_names=class_labels))

    # Create confusion matrix
    conf_matrix = confusion_matrix(y_true, y_pred)

    # Plotting confusion matrix heatmap
    plt.figure(figsize=(8, 6))
    sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
                xticklabels=class_labels, yticklabels=class_labels)
    plt.title('Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()
    print("\n\n")

# Print evaluation for each model

models = {"LeNet": lenet, "Alex Net": alex_net, "VGG-16": vgg16_net, "GoogleNet": googlenet, "ResNet": resnet}

for model_name, model in models.items():
    print_evaluation(model, validation_generator, model_name)
```

FIGURE 17: Model Performance Comparison

5.3.3 Project Deployment

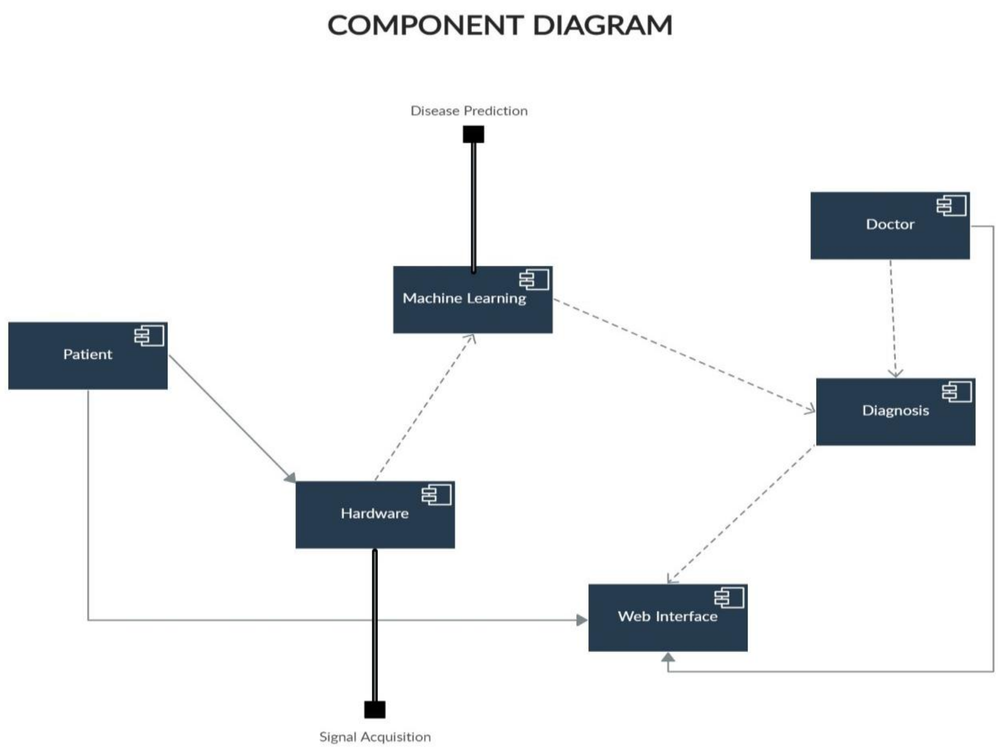

FIGURE 18: Project Deployment

5.3.4 System Screenshots

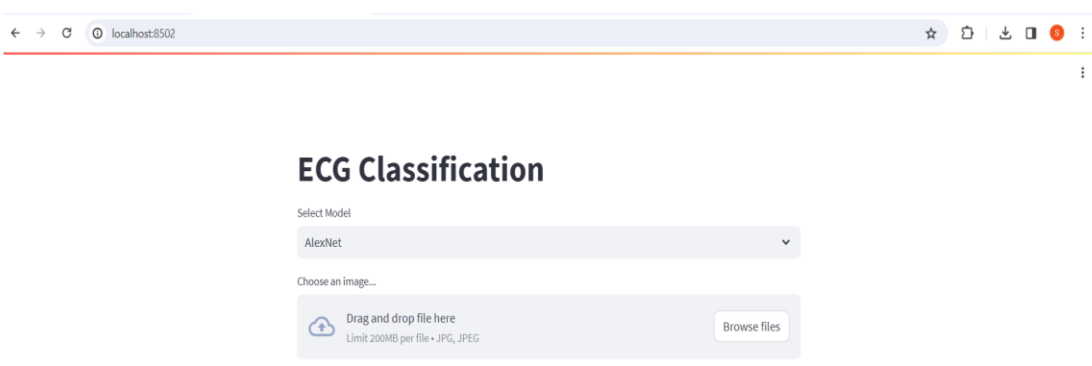

FIGURE 19: Web App Deployment

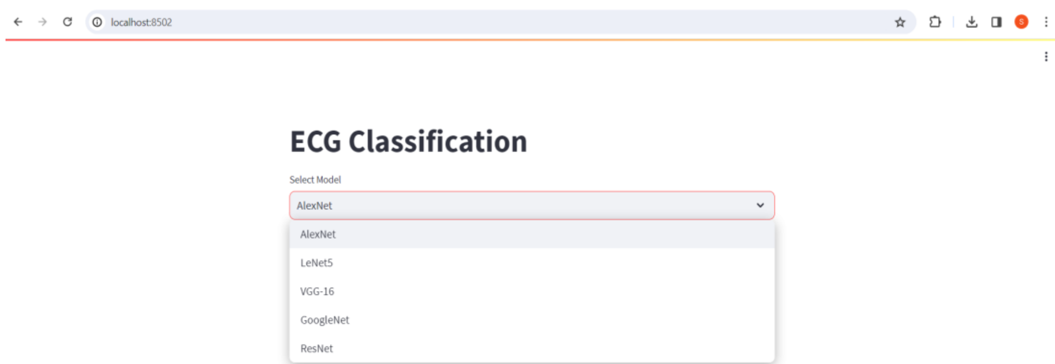

FIGURE 19: Web App Deployment

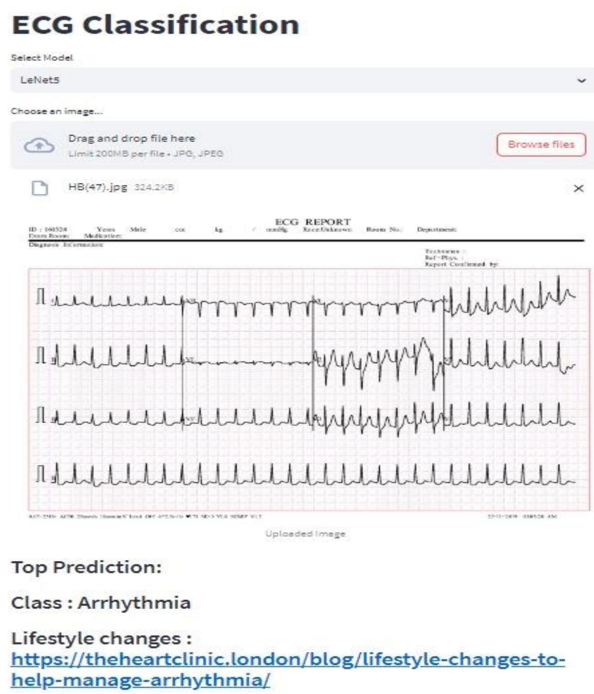

FIGURE 20: Web App Deployment- Results

5.4 Testing Process

5.4.1 Test Plan

Objective: Ensure that all features in our application function according to specifications.

Scope: Test all software components interacting with the application.

Approach: Implement an approach wherein each component is tested separately and then after integration with the main application, it should also include user experience testing.

Focus: Emphasize the development of a user-friendly app and interface, ensuring easy interaction and a hassle-free experience for the user.

5.4.2 Features to be Tested

The primary goal of the project was to determine the patient's cardiac state from the recorded ECG signals. These features included the following:

ECG Signal Image Acquisition (Hardware Testing): Validating the hardware's capability to accurately capture and process ECG signal images for subsequent analysis.

ECG Signal Image Classification: Assessing the accuracy of the image classification process for disease prediction.

5.4.3 Test Strategy

This project’s testing strategy includes specific steps to ensure perfect performance. Initially, we will test the hardware’s ability to record and transmit ECG signal images. Next, extensive testing of imaging algorithms will assess their accuracy in preparing images for disease prognosis and lifestyle recommendations. Validation against dataset using five different models will provide the accuracy and reliability of disease prognoses have been ensured. Also, the testing in the web application focuses on enabling users to use any selected model with selected images, confirming seamless integration and functionality. Throughout testing, emphasis will be on accuracy, reliability, and seamless interaction between hardware, image processing, multiple models, and user interface within the web application, ensuring optimal performance in disease prediction and lifestyle recommendation functionalities.

5.4.4 Test Techniques

In this project, each component was tested on its own and then tested how they work together. First, it was verified that the ECG signal capture device was working properly. Next, it examined how images from the device were processed to diagnose diseases and predict lifestyle changes. Five different prediction methods were tested on these images to see which performed best. This website allows us to choose any style to use in the image. Everything was tested to see if it all worked well together and provided accurate disease predictions.

5.4.5 Test Cases and Test Results

1) Hardware Functionality Test:

Test Case 1: Signal Recording

1) Input: Record ECG signals using the hardware.
2) Expected Output: Successful and accurate recording of ECG signals.

2) Model Testing:

Test Case 2: Model Predictions

1) Input: Apply five different models to processed images for disease prediction.
2) Expected Output: Obtain accurate disease predictions from diverse models.

3) Web Application Functionality:

Test Case 3: Model Selection

1) Input: Choose any model for prediction on a selected image within the web app.
2) Expected Output: Successful application of the chosen model on the image.

5.5 Results and Discussions

As we can see , our model has significantly learned from the training dataset and reduced the training loss by a significant level.

LeNet

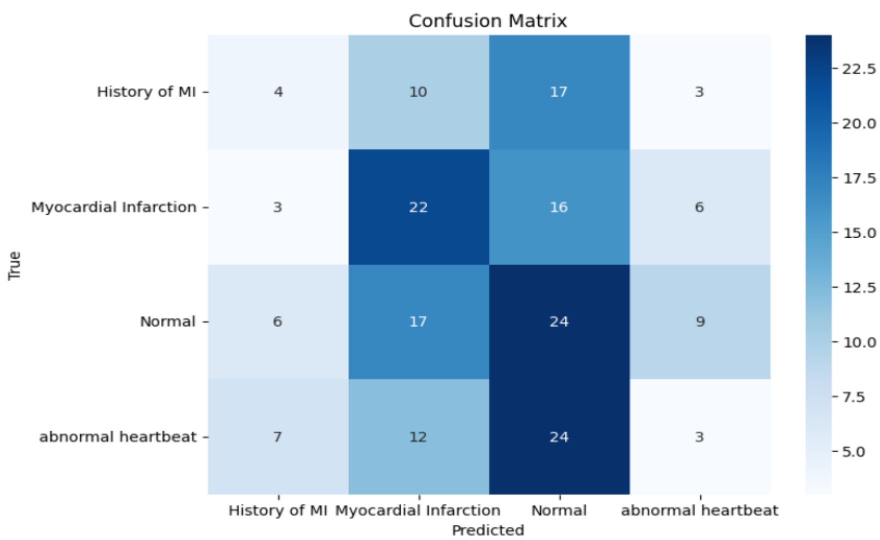

FIGURE 21: Confusion Matrix (LeNet)

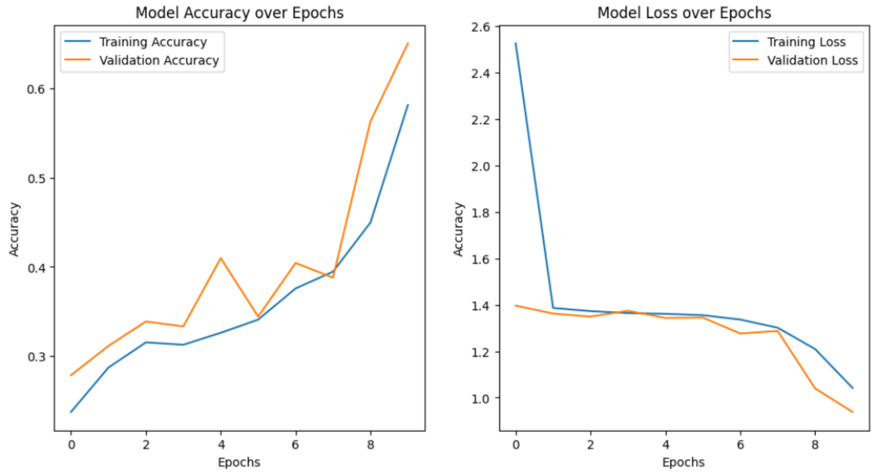

FIGURE 22: Model Accuracy and Loss over Epochs(LeNet)

ResNet

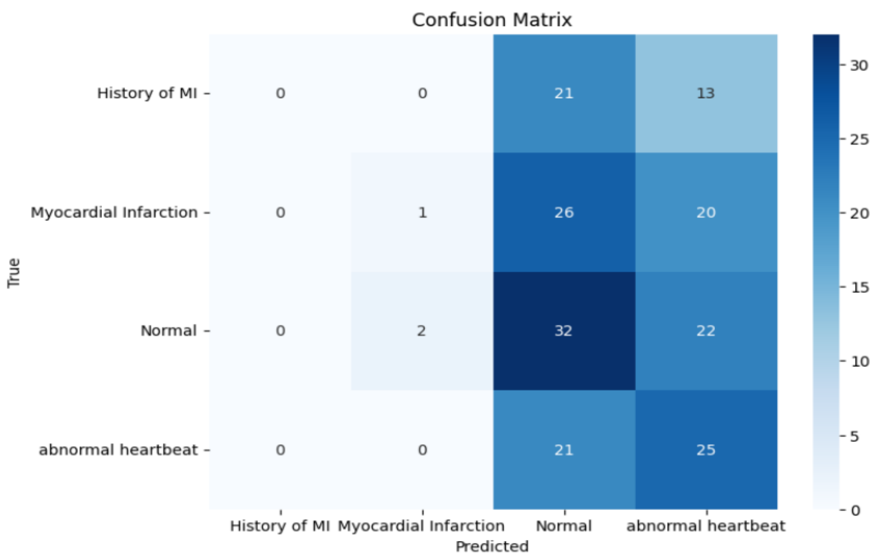

FIGURE 23: Confusion Matrix(ResNet50)

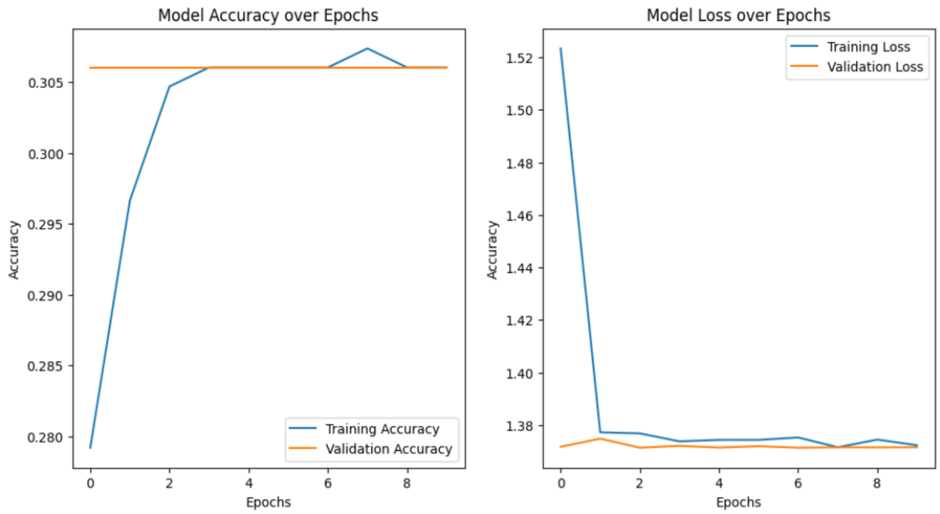

FIGURE 24: Model Accuracy and Loss over Epochs(ResNet50)

5.6 Inferences Drawn

Based on the results of the test cases we can conclude that our app is made as per the specifications and is operating as intended. Testing data for the deep learning model suggests that they are well trained and fulfill the requirements of the project. Our deep learning model has a few categories. Hence in our future endeavors we will try to design a deep learning model which can process more categories of data.

5.7 Validation of Objectives

Table 8: Objectives

S No.	**Objective**	**Status**
1	Designing an Arduino based ECG Machine using AD8232 ECG Sensor	Successful:Multiple ECG Signals were recorded
2	Creation of webapp and deployment of ML model in it	Successful: Detecting arrhythmia , myocardial infarction and normal patients successfully.
3	Providing the patients with lifestyle changes.	Successful

CONCLUSIONS AND FUTURE DIRECTIONS

6.1 Conclusions

This report presents a substantial corpus of work. The precision and integration parts of the process presented some early difficulties, but our team eventually collaborated to create the gadget. The Cardiologist Assistant is a noteworthy advancement in the use of cutting-edge technology to improve cardiac treatment. This project's successful development has demonstrated that it is feasible to construct a portable, reasonably priced monitoring device that provides real-time insights into a person's cardiovascular health. This assistant can detect irregularities and anomalies that might otherwise go unnoticed during routine medical checkups by monitoring metrics like heart rate and ECG.

Because of its real-time data analysis capabilities, cardiac problems can be identified early, allowing for prompt medical intervention and individualized management plans. Crucially, by giving patients access to their health information, this approach encourages them to take an active role in their cardiac health by helping them make educated decisions regarding treatment adherence and lifestyle modifications. The assistant can also be used for virtual consultations and to lower geographical obstacles to high-quality healthcare thanks to its remote monitoring capability. Its invention has been greatly aided by the cooperation of cardiology and technology specialists, highlighting the complementary roles that these fields play in advancing medical solutions. As we celebrate these successes, we also need to recognize that sensitive medical data carries security and privacy risks, which emphasizes the importance of strong data protection protocols. The initiative also recognizes its shortcomings, such as possible measurement errors and its applicability to a range of patient populations.

In the long run, the cost-effectiveness of this assistant points to a viable solution for healthcare environments with limited resources. This Arduino-based cardiology assistant is a useful tool that can also be used as a teaching tool. It shows how electronics and healthcare can be used to improve patient outcomes. Its accuracy must be improved and its capabilities must be expanded through continuous research and development, which will usher in a new era of patient-centric cardiac care.

6.2 Environmental/ Social/ Economic Benefits

The Cardiologist Assistant brings about a triple win for the environment, society, and economy, bringing about a revolution in healthcare.

Regarding the environment, its small size and constant monitoring capabilities greatly lessen the environmental impact of conventional medical procedures. This helper reduces the amount of resource-intensive medical equipment, which lowers energy use and related carbon emissions and promotes a more sustainable healthcare ecosystem.

From a societal standpoint, It promotes patient participation and educated decision-making about lifestyle selections and treatment adherence by offering real-time data and customised insights. This empowerment improves well-being, encourages better practices, and may reduce the incidence of heart illnesses.

From an economic standpoint, the Arduino-based assistant can save a significant amount of money across the healthcare system. Because of its economical design, advanced cardiac monitoring technology is now more widely accessible to patients and healthcare institutions. Early diagnosis of cardiac problems and prompt intervention are made possible, which lowers the need for hospital stays, intense treatments, and emergency care and, in the end, lowers healthcare costs. The project's partnership between technologists and cardiologists also promotes multidisciplinary innovation, which could spur economic growth by facilitating the development and commercialization of related healthcare products.

When taken as a whole, these advantages transform the healthcare environment. Along with improving patient outcomes, the assistant promotes social justice, environmental sustainability, and economic resilience. The Arduino-based cardiology assistant represents the possibility of combining technology and medicine to promote a healthier, more inclusive, and environmentally conscious society by forging a route towards patient-centric, eco-conscious healthcare solutions. Future developments in healthcare appear promising as long as these ideas are investigated, improved, and put into practice. In this scenario, holistic well-being will coexist peacefully with economic responsibility and environmental stewardship.

6.3 Reflections

As we approach the completion of this project, looking back, working on it was a challenging task that pushed us to expand our knowledge beyond what we learned in the course curriculum. It required the support and collaboration of all team members, and together, we achieved the objectives of our project. We saw it as an opportunity to understand new things and gain invaluable knowledge in various technical domains. This experience also taught us how to work collaboratively as a team to achieve the required outcomes. We are grateful for the opportunity to work on a project that challenged us and forced us to push our boundaries.

6.4 Future Work

Enhancing Accuracy: Work on refining and improving the accuracy of disease prediction and lifestyle recommendations by fine-tuning the models, incorporating more diverse datasets, or exploring advanced algorithms.

Collaboration with Hospitals: Collaborate with healthcare institutions or hospitals to test and integrate your system into their healthcare processes. This could involve clinical trials or real-world testing to validate the system's effectiveness in a medical environment.

Regulatory Compliance: Ensure compliance with healthcare regulations and standards if considering deployment in clinical settings. This might involve obtaining necessary certifications or approvals.

7.1 Challenges Faced

Lack of datasets: Comprehensive ECG records for each of the three heart diseases were either scarce or nonexistent in the internet databases that were accessible. There was one such dataset that had exclusively pictures of a specific illness. The dataset had to be built by integrating several datasets for each cardiac condition into a single labelled dataset, though, because we did not want to restrict the model. It took a lot of investigation to find the appropriate sources for choosing a high-quality dataset for a precise model.

CNN Attempt: An effort was made to train a CNN model on the collected dataset for the purpose of categorizing the identified ailment. When the dataset was trained using the CNN model, the following outcomes were attained: 47.8%. We had to go far and wide for alternative learning strategies because of the low validation accuracy.

Size of dataset: There was a restriction to the size of the disease detection dataset because it was gathered from several sources with restricted access to patient records. There weren't many patient records gathered. As a result, in order to boost the dataset's size and improve the model's accuracy, we had to employ picture augmentation.

Unbalanced Data: Each training heart condition folder included unequal photos from the illness detection dataset. Since an imbalanced dataset significantly affects the accuracy of the model, we included a code snippet to add balanced classes.

7.2 Relevant Subjects

Table 9: Subjects

Subject Code	Subject Name	Description
UML501 and UCS761	Machine Learning And Deep Learning	Helped us in implementing and designing our machine learning and deep learning models.
UCS503	Software Engineering	Helped us in designing our project using UML diagrams.

7.3 Interdisciplinary Knowledge Sharing

The team has acquired a great deal of information and expertise from various sources in addition to applying the knowledge we have learned in our relevant courses from the course curriculum, all of which are necessary to finish this project. Concepts from software engineering were helpful in composing the report and creating the project's diagrams. We also used advanced deep learning models which were taught to us in deep learning (UCS761)

7.4 Peer Assessment Matrix

Table 10: Peer Assessment Matrix

	Evaluation of					
Evaluation by	**Name of Team Members**	Twesha Arvind	Ishita Kaundal	Parv Gupta	Shantam Anand	Varchasva Singh
	Twesha Arvind	NA	**5**	**4.5**	**4.5**	**4.5**
	Ishita Kaundal	**4.5**	NA	**5**	**4.5**	**4.5**
	Parv Gupta	**4.5**	**4.5**	NA	**5**	**4.5**
	Shantam Anand	**4.5**	**4.5**	**4.5**	NA	**5**

	Varchasva Singh	**5**	**4.5**	**4.5**	**4.5**	NA

7.5 Role Playing and Work Schedule

Table 11: Role Playing and Work Schedule

S.No.	**Name**	**Role**
1.	Twesha Arvind	Research of existing techniques/research papers for the entire ML/DL Framework. Deployment of webapp using streamlit Documentation of the project report Studied and applied Alex Net deep learning model
2.	Ishita Kaundal	Research of existing techniques/research papers for the entire ML/DL Framework. Studied and applied LeNet deep learning model Deployment of webapp using streamlit Documentation of the project report
3.	Parv Gupta	Research of existing techniques/research papers for the entire ML/DL Framework. Studied and applied VGG-16 deep learning model Implementation of hardware using Arduino Nano and AD8232 ECG Sensor Documentation of the project report
4.	Shantam Anand	Research of existing techniques/research papers for the entire ML/DL Framework. Studied and applied GoogleNet deep learning model Deployment of webapp using streamlit Documentation of the project report

5.	Varchasva Singh	Research of existing techniques/research papers for the entire ML/DL Framework. Implementation of hardware using Arduino Nano and AD8232 ECG Sensor Studied and applied ResNet-50 deep learning model Documentation of the project report

7.6 Student Outcomes Description

Table 12: Student Outcomes

Sno.	**DESCRIPTION**	**OUTCOME**
1.1	Ability to identify and formulate problems related to computational domain	Identified an ML based problem of classifying the signals generated by the ECG hardware and later classifying the signal images in various heart conditions and providing lifestyle changes along with it.
1.2	Apply engineering, science, and mathematics body of knowledge to obtain analytical, numerical, and statistical solutions to solve engineering problems.	Applied machine learning and deep learning to solve problems related to the classifying the signal images in various heart conditions and providing lifestyle changes along with it.
1.3	Analyse and interpret results with respect to assumptions, restrictions and theory.	Build a website and deployed various ML/DL model in it. The website will provide user with lifestyle changes to regulate back to normalcy.
1.4	Prepare and present variety of documents such as project or laboratory reports according to computing standards and protocols.	Prepared reports for capstone evaluations using UML diagrams and the given report format.
1.5	Able to communicate effectively with peers in well organized and logical manner using adequate technical knowledge to solve computational domain problems and issues.	Communicated with peers effectively and were able to deliver the final application
1.6	Participate in the development and selection of	Developed a web application in

	ideas to meet established objective and goals.	python using streamlit over multiple iterations selecting multiple libraries and frameworks as needed
1.7	Able to plan, share and execute task responsibilities to function effectively by creating collaborative and inclusive environment in a team	Worked together with a team to deliver the project.
1.8	Ability to perform experimentations and further analyze the obtained results.	Analysed and tested various machine learning and deep learning models to address sub-problems
1.9	Ability to analyze and interpret data, make necessary judgements and draw conclusions.	Were able to select appropriate libraries, models and algorithms according to the need of the problem
2.0	Able to explore and utilize resources to enhance self-learning	Learned on our own how to develop website in python using streamlit and how to use transfer learning in ResNet to classify images
2.1	Apply various codes and algorithmic processes.	Various codes and conversions were used while writing the code
2.2	Deliver a clear and effective oral presentation.	The team delivered PPTs on various occasions to show the progress to our mentor and the panel.

7.7 Brief Analytical Assessment

Q1. From what sources of data did your team gather information to compile a list of possible project problems?

Answer: The team members were aware of the project's requirements as well as a few issues that needed to be to be investigated. We investigated the current systems in use, evaluated their shortcomings, and flaws by examining several IEEE technical periodicals and journals. The challenges of integration of the ML/DL model and software were verified again using the formal records of the technology in use. With the help of our mentor, the project's scope was further determined.

Q2. How did your project team solve the difficulties in the project using analytical, computational, and/or experimental methods?

Answer: To find answers, the analytical techniques included conducting a lead generating survey, a review of the literature to analyze current frameworks, find research gaps, and conduct studies current remedies. UML and design diagrams, as well as the software requirement specification were equally crucial to consider potential solutions and implement the required adjustments. Experimental techniques such as prototyping, analyzing it, and determining the effectiveness of the product through testing contributed to the ultimate solution.

Q3. Did the project need proof of basic understanding of scientific and/or technical principles? How did you apply, if you did?

Answer: It's true that the project required proof of basic scientific and/or engineering concepts. The principles of several topics, including machine learning, had to be applied during the development and integration of our program, such as software engineering, etc. Additionally, the tenets of our project was thoroughly documented using software engineering.

Q4. During the project, what resources did you use to gain new information that was not covered in class?

Answer: To learn more about the subjects we are working on, the team used the internet and a variety of online tutorials. Numerous YouTube classes teaching how to create different models and their execution were cited, as well as numerous articles and technical reports on IEEE to discover fresh information not covered in class. additionally cited handbooks available on the websites of the models in use and put into action.

Q5. Does the project help you understand how important it is to use engineering to address problems in the real world and might the project development help you become adept with software development environments and tools?

Answer: The project uses engineering to solve a practical issue. Working on this project has enabled The group has inspired us to take on new challenges by realizing the importance of solving problems in the real world in a variety of fields. The group was exposed to a number of novel technologies, including deep learning, building websites, etc.

REFERENCES

[1] F. Corradi et al., "ECG-based Heartbeat Classification in Neuromorphic Hardware," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019.

[2] Ertan Butun, Ozal Yildirim, Muhammed Talo, Ru-San Tan, U. Rajendra Acharya, "1D-CADCapsNet: One dimensional deep capsule networks for coronary artery disease detection using ECG signals", Version of Record (2020)

[3] Shwet Ketu and Pramod Kumar Mishra, "Empirical Analysis of Machine Learning Algorithms on Imbalance Electrocardiogram Based Arrhythmia Dataset for Heart Disease Detection", Arabian Journal for Science and Engineering volume 47, pages1447–1469 (2022)

[4] Saira Aziz, Sajid Ahmed and Mohamed-Slim Alouini, **"**ECG-based machine-learning algorithms for heartbeat classification", Scientific Reports volume 11, Article number: 18738 (2021)

[5] Dr. T. Vijayakumar, Mr. R. Vinothkanna and Dr. M. Duraipandian, "Fusion based Feature Extraction Analysis of ECG Signal Interpretation – A Systematic Approach", Journal of Artificial Intelligence and Capsule Networks (2021) Vol.03/ No.01